DTHEX: AN IMPROVED ALL-HEXAHEDRAL MESHING SCHEME USING GENERAL COARSENING TOOLS

by

Sara L. Richards

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Civil and Environmental Engineering

Brigham Young University

December 2003

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Sara L. Richards

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date	Steven E. Benzley, Chair
Date	Steven J. Owen
Date	Henry N. Christiansen

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Sara L. Richards in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date

Steven E. Benzley
Chair, Graduate Committee

Accepted for the Department

Wayne C. Downs
Graduate Advisor, Department of Civil and Environmental Engineering

Accepted for the College

Douglas M. Chabries
Dean, College of Engineering and Technology

ABSTRACT

DTHEX: AN IMPROVED ALL-HEXAHEDRAL MESHING SCHEME USING GENERAL COARSENING TOOLS

Sara L. Richards

Department of Civil and Environmental Engineering

Master of Science

Certain applications of the finite element method require hexahedral meshes for the underlying discretization. A procedure guaranteed to produce an all-hex mesh is to begin with a tetrahedral mesh and then subdivide each element into four hexahedra. This procedure, referred to as THexing has certain advantages and disadvantages, which will be presented in this thesis. While the approach is not ideal for many models, it is currently the only viable hex-meshing alternative that can be used on arbitrary geometries. Because it is so widely used, there is great interest in improving the quality of resulting meshes, even if in small degree. This research presents a method for improving the THex approach, known as Diced THexing, or DTHexing.

The DTHex approach is based on general coarsening tools. An initial triangle surface mesh is coarsened and smoothed iteratively until a coarse mesh of reasonable quality is obtained. The volume is then easily meshed using a tetrahedral scheme. Each tetrahedron is subdivided into four hexahedral elements, and then each hexahedron is further subdivided into eight hexahedra. The goal of this method is to 1)

improve the quality of elements in the finite element mesh and 2) decrease the number of overall nodes. Examples will be shown that highlight the improvement in each of these areas.

The research presented is particularly useful in the fields of biomechanics and biomedical engineering. Medical scanning techniques typically produce triangular meshes that are very fine, and further subdivision makes analysis infeasible with current computing power. The DTHex approach has been successful at improving these meshes without decreasing element size.

ACKNOWLEDGMENTS

I would like to give many thanks to my committee members for their time and effort in assisting me in this work. I would also like to thank Jason Shepherd of Sandia National Labs for providing the funding and motivation for this research and Andy Anderson and Jim Weiss from the University of Utah for providing the biological models.

Contents

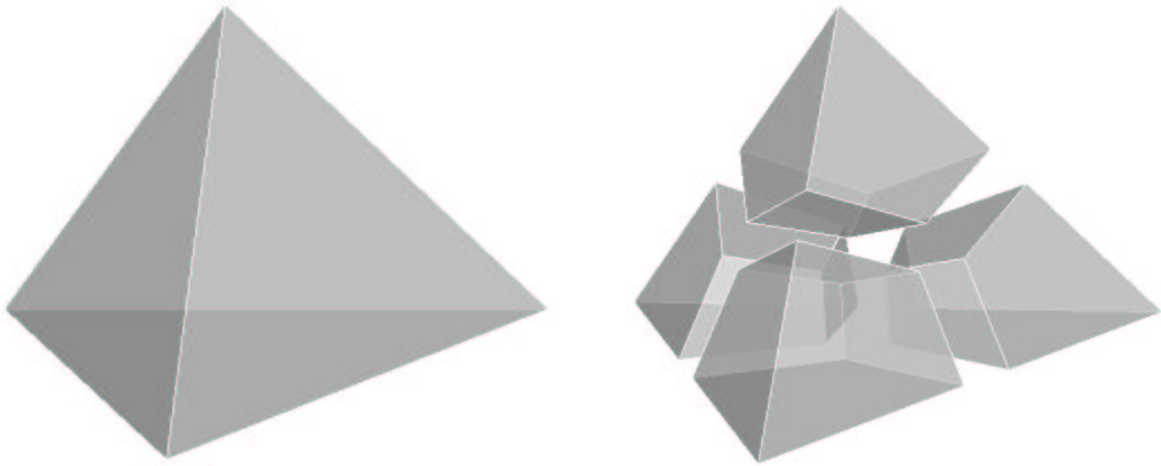
List of Tables

List of Figures

1 INTRODUCTION

Finite element analysis is a numerical approach for analyzing multivariable systems with piece-wise approximations. Although the first applications of the finite element method were limited, it is now widely incorporated in many disciplines such as structural engineering, computational fluid dynamics, microelectronics, groundwater flow, aerodynamics, computational medicine, mechanical engineering, and electrical engineering. As computing power continuously increases, the type and complexity of problems that can be solved also increases.

Because finite element analysis is a numerical procedure, its success is closely tied to the accuracy of the discretization (i.e. the mesh). While it would be ideal to use extremely fine meshes in all cases, it is not computationally feasible to perform analyses on such refined models. Rather, the analyst must try to get the "best mesh possible" for the available computing resources. Some analysis codes have been written to work only with hexahedral elements, since it has been shown that hexahedral elements have some desirable qualities that allow them to perform better than their tetrahedral counterparts for a given number of degrees of freedom [2][3][7][8][38]. Since the early 1990's, research has produced several methods for producing hexahedral meshes, which will be discussed in Chapter 2. However, none of these methods produces the "best possible" mesh in all situations, and significant user intervention is still required to produce acceptable meshes. To date, the only method that is guaranteed to generate all-hexahedral meshes on arbitrary geometries is known as THexing[30]. THexing is the process of splitting each element in a tetrahedral finite element mesh into four hexahedral finite elements as shown in Figure 1.1.



a. b.

Figure 1.1: THexing a single tet.

One of the newer applications of the finite element method for three dimensional problems is in biological modeling. Initial biological meshes are often processed using scanning and imaging devices and software. These meshes typically have a single topological surface, a large number of features, regions of high curvature, holes, long narrow filaments, and a large number of nodes. In addition, there are often many inaccuracies and inconsistencies in the mesh that are generated in the data transfer process which require significant modification before meshing is possible. In almost all cases, the mesh is created as a triangle surface mesh. Tetrahedral meshing is then possible for the volume, using existing meshing technology[12][13]. Once a valid tetrahedral mesh is created, it can be THexed to form an all-hexahedral mesh. THexing is a simple process, but often generates more nodes that can be accomodated by the analysis code. Even though THexing is not ideal, it is the only fully automatic method currently available for meshing most biological models. Figures 1.2 and 1.3 show two biological meshes (used with permission from researchers at the University of Utah biomedical department). This research explores the possibilities of improving the THexing approach. We present an algorithm known as DTHex that is based on general coarsening and refinement procedures. The goal of this research is to develop a general approach for meshing complex models that produces elements with better quality than THexed meshes while decreasing the number of nodes for analysis to thousands instead of millions.

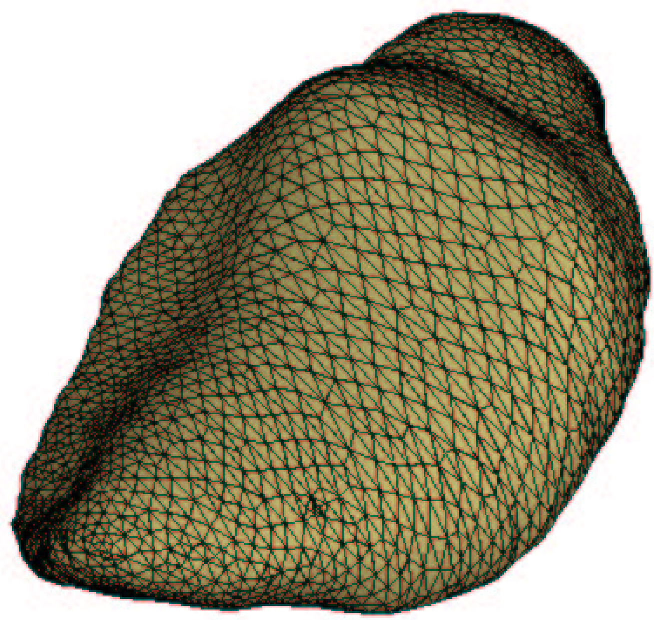


Figure 1.2: A spheroidal biological model.

This thesis is organized as follows. Chapter 2 is a survey of existing hex-meshing technology. This chapter is not meant to be an all-encompassing list, but rather to highlight some of the approaches that researchers have used and to describe the advantages and disadvantages of each method. This chapter also discusses some of the general challenges associated with producing conformal hexahedral meshes. Chapter 3 introduces the DTHex approach and provides a general overview to the method. Chapters 4, 5, 6 discuss the details of each step in the algorithm. Examples are given in Chapter 7. Finally, Chapter 8 presents some concluding thoughts and

Figure 1.3: A complex biological model.

directions for future work, as well as summarizing the impact of the research. Chapter 9 is a glossary of meshing terms that will be referenced throughout the rest of the work. Readers who are unfamiliar with meshing terminology are advised to review this chapter.

2 HEXAHEDRAL MESHING

For many years, the solution to the automatic hex-meshing problem has been considered a "holy grail" in the analysis community[3]. Hexahedral meshing is challenging because of the layered nature of hexahedral mesh sheets. This thesis does not attempt to present a complete solution to the automatic hexahedral mesh generation problem, but rather an improvement to current techniques. In light of that goal, this chapter introduces some of the challenges associated with automatic hexahedral mesh generation, and then briefly discusses the different approaches to the problem.

2.1 Hexahedral Meshing Challenges

Hexahedral elements are desirable in finite element analysis because they allow for directionality in the mesh. In addition, they may capture the geometry more accurately, decrease the overall element count, and converge faster than tetrahedral meshes.[3] For these reasons, many of the analysis codes are written to work with hexahedral elements instead of tetrahedral elements. It is relatively trivial to create a tetrahedral mesh on any geometry, but it is much more difficult to create valid hexahedral meshes on arbitrary geometries.

The greatest challenge in hexahedral meshing is the inability to have local modification. Every face on a hexahedron has an opposite face which is connected to another hex. One may continue following this layer of elements or dual chord, connecting each hex to a neighboring hex, until it either reaches a boundary or connects back on itself. Thus whenever an element is added to the mesh, an entire sheet of elements must be added. In tetrahedral meshing this problem is avoided because elements do not compose layers; thus, localized modification is much easier.

Another problem faced in hexahedral meshing is the small angle problem[3]. Every volume must admit a quadrilateral mesh on its surface. If there are any small angles on the surface, they will form very poor quadrilaterals when meshed. See Figure 2.1.

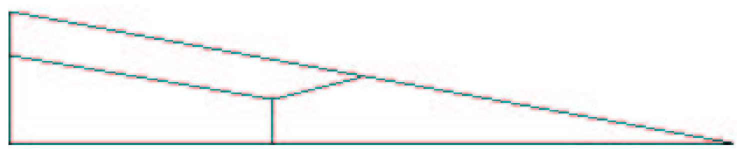


Figure 2.1: A poor quadrilateral mesh formed from small surface angles.

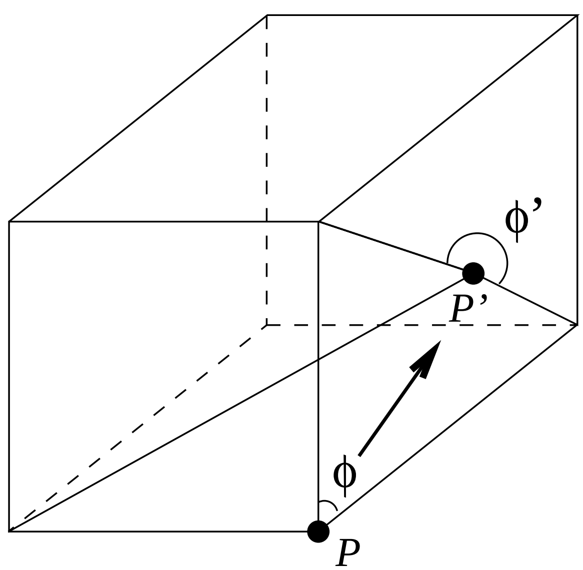


Figure 2.2: Inverting a hexahedral element.

One final problem is that hexahedrons may become concave if any of the interior angles between adjacent faces exceeds π. This is known as an inverted element, or negative Jacobian element. Inverted elements present a problem during finite element analysis. Tetrahedral elements do not present as much of a problem because a tetrahedral element cannot become inverted until one of its node passes the plane of its other three nodes. A hexahedral mesh may become inverted before it crosses the plane of the opposite face. Figure 2.2 shows an example of an inverted element.

A fully robust automatic hexahedral meshing scheme should be able to meet the following criteria:

- Element Quality - The mesh should have elements with reasonable quality and should not produce inverted elements.
- Element Uniformity- The mesh should be composed entirely of hexahedral elements, and quality should not vary widely from element to element.
- Geometric Generality - The algorithm should work on arbitrary geometries
- Boundary Sensitivity - Elements on the boundary should roughly follow the contours of the boundary.
- Orientation Insensitivity - The orientation of the geometry should not affect the mesh
- Size Control - The user should be able to control element size
- Speed - The algorithm must generate millions of elements in reasonable time
- User Intervention - The algorithm should be automated, requiring minimal user intervention

The following sections discuss existing techniques and how well they meet these criteria.

2.2 Sweeping and Multi-sweeping

In sweeping and multi-sweeping a geometry is decomposed into topological cylinders made of topologically similar source and target surfaces, that are connected by mapped linking surfaces[4][18][25]. In sweeping, there is only one source and target surface, while in multi-sweeping there are multiple source and target surfaces. Research continues on methods of automating the decomposition process[20], but the current approach involves significant user intervention. Figure 2.3 shows a multi-swept mesh. The advantages of the sweeping and multi-sweeping algorithms are:

- Good quality elements
- Good boundary elements
- Speed of algorithm (not including the time required for user decomposition)

The disadvantages of the sweeping and multi-sweeping algorithms are:

- Poor geometric generality
- Poor size control
- Significant time required for decomposition

2.3 Indirect Hexahedral Meshing

Indirect hexahedral mesh generation involves decomposing a geometry into tetrahedra and then applying templates or transformations to the tetrahedra to form either an all-hexahedral mesh or a hex-dominant mesh.

2.3.1 THexing

The most common form of indirect hexahedral meshing is to mesh the volume with tetrahedra and then split each tetrahedra into four hexes, as introduced in Chapter 1. Because it is so easy to implement, this procedure is used quite often to produce all-hexahedral meshes. The technique has the following advantages:

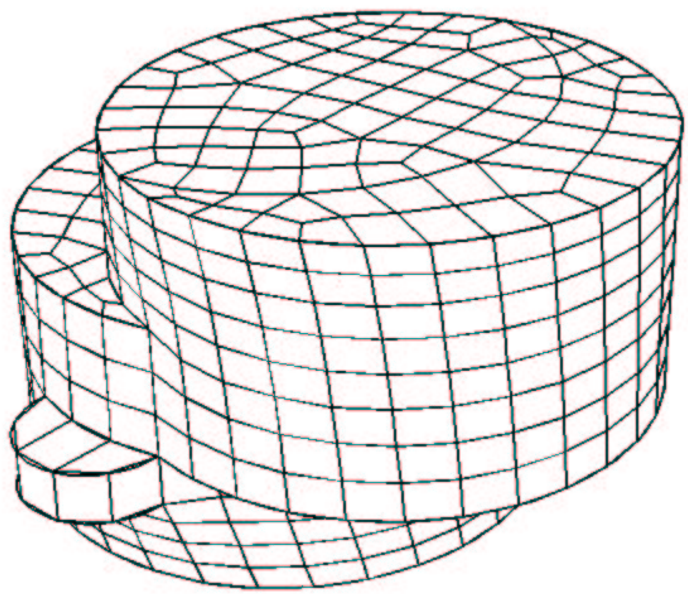


Figure 2.3: An example of a multi-swept mesh.

- Speed
- Requires no user intervention
- Geometric generality
- Orientation insensitivity
- Size control based on tet mesh

THexing has the following disadvantages:

- Poor boundary sensitivity
- Increased element count
- Poor element quality
- Non-uniform element quality

2.3.2 HMorph

Another approach to indirect hexahedral mesh generation was proposed by Owen [28]. This method also starts with an initial tetrahedral mesh, but instead of decomposing the tets, it goes through a series of transformations to combine the tets to form a hexahedral mesh. The process begins at the surface, and proceeds inward until no more tets can be combined into hexes. The final mesh is then a hybrid of hexahedra, pyramids, and tetrahedra. Advantages of the H-Morph algorithm:

- Boundary sensitivity
- Geometric generality
- Orientation insensitivity
- Size Control

Disadvantages of the algorithm are:

- Leaves some non-hexahedral elements in the mesh
- Poor element quality on non-hexahedral elements

2.3.3 PolyFEM

Meshkat and Talmor[23] propose a method similar to the HMorph technique which is implemented in the PolyFEM software. Their technique also begins with an initial tetrahedral mesh. Some of the tetrahedra are combined into hexahedra using a generalized graph representation. All possible tetrahedral compositions of pentahedra and hexahedra are found using the graph representation, and the algorithm searches through the tetrahedral subspace to find subgraphs which can be formed into hexahedral elements. The final mesh is a hybrid mesh of hexahedra, pentahedra, and tetrahedra. A main advantage of HMorph technique over Polyfem is that HMorph is boundary sensitive, but Polyfem is not. HMorph generally can create 90 percent hexes by volume. PolyFem is typically less than 50 percent.

2.3.4 HEXHOOP

In order to enforce an all-hexahedral mesh on an otherwise hex-dominant mesh, Yamakawa and Shimada propose an extension of an indirect approach, known as HEXHOOP [41]. The HEXHOOP algorithm is a set of mesh conversion templates that automate the conversion of a hex-dominant mesh to an all-hex mesh. An initial hybrid mesh, defined using a method similar to Meshkat and Talmor [23] is given of hexahedra, pentahedra, and tetrahedra. A set of patterns are applied to the mesh which allow the mesh to be refined into an all-hex mesh. One such pattern is the conversion of a tetrahedron into four hexahedra. Advantages of the HEXHOOP algorithm are:

- All-hexahedral mesh is guaranteed
- Geometric Generality
- Speed
- Good element size in non-transition regions

Disadvantages of the algorithm are:

- Element quality is poor on some transition elements.
- Element quality is not uniform
- Poor element size control in transition regions

2.4 Grid Overlay

Another class of hexahedral mesh generation is known as the grid overlay method[32], which is also referred to as an "inside-out" method. A bounding box or overlay grid is created around the model; then the box is meshed to a desired fineness. All of the elements that are outside of the box are discarded, and the remaining elements are smoothed and reoriented to fit the boundary. Although it is a straightforward technique, it becomes very difficult to perform the last step of fitting the boundary. The boundary elements have much poorer quality than the rest of the mesh. In addition, the orientation of the bounding box determines the final mesh. Advantages of the grid overlay method are:

- All hexahedral mesh is created
- Good element quality on interior elements

Disadvantages of the grid overlay method are:

- Element size must be uniform
- Poor element quality on boundary elements
- Poor orientation sensitivity
- Poor geometric generality

2.5 Plastering

Plastering is a three-dimensional advancing front algorithm. An advancing front algorithm is one that starts at the boundary of a volume and progressively adds layers of elements as it progresses to the interior. As these advancing fronts collide, modification templates are applied to produce a conformal mesh. As one might expect, this produces good elements that match the contours of the topology on the boundary, but leaves poor elements in the interior. Many times, the algorithm is not able to successfully combine all of the fronts, leaving skewed and wedge-like elements inside the mesh. Further research is still necessary to create a fully robust plastering scheme. Current advantages of the approach are:

- Geometric Generality
- Boundary Sensitivity
- Orientation Insensitivity
- Size Control

Disadvantages of the plastering approach are:

- Cannot yet guarantee an all-hexahedral mesh
- Poor quality elements at front collisions
- Speed

2.6 Whisker Weaving

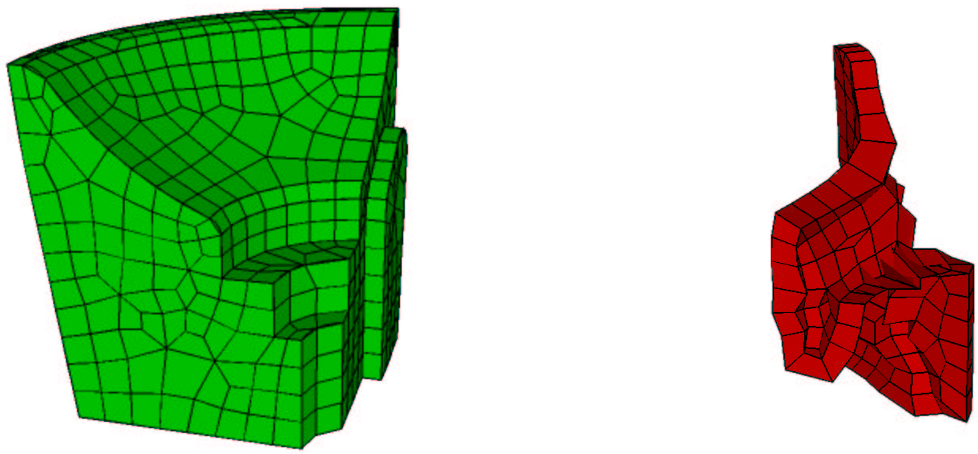
Whisker Weaving is an unstructred hexahedral meshing scheme introduced by Tautges, Blacker, and Mitchell [36] based on the principles of the dual that are introduced in Chapter 9. An initial quadrilateral surface mesh is given. Chords or whiskers are projected to the interior of the volume from the centroids of the surface quadrilaterals. This continues until all of the chords have connected with other chords to form the dual of the mesh. A series of cleanup procedures is applied, and then the hexahedral mesh is extracted from the dual. Whisker weaving is another algorithm that has great potential but is still in research and development stages. Current advantages of the whisker weaving approach are:

- Guarantees an all-hexahedral mesh
- Boundary sensitivity
- Orientation insensitivity
- Size control on surface mesh
- Geometric generality

Disadvantages of the whisker weaving algorithm are:

- Inverted elements
- Element quality is not uniform
- Speed

Figure 2.4a. shows a mesh created from whisker weaving. Figure 2.4b. shows one of the dual sheets of the mesh.



a. b.

Figure 2.4: Example dual sheet from a mesh created with whisker weaving.

3 OVERVIEW OF ALGORITHM

DTHex, which stands for "Diced THexing" is an improved method for generating all-hexahedral mesh elements using an indirect hex-meshing approach. This chapter gives an overview of the DTHex method for all-hexahedral mesh generation.

The discussion in Chapter 2 describes the current approaches to hexahedral meshing. It should be clear that robust hexahedral meshing for arbitrary volumes is not a solved problem. This research is not intended to be a solution to the all-hexahedral meshing problem for all classes of geometry, but proposes a solution to a specialized problem, namely hexahedral meshing for biological models. The focus is on improving the THex approach, since that method is the most commonly used for complex models, and has a great potential for impact in this domain. As an example of the poor quality of THexed meshes, Figures 3.1a. and 3.1b. show the dual sheets of a model that has been THexed. The goal in DTHex is to manipulate the dual sheets so they are more regular, which can lead to better quality elements. Figures 3.1c. and 3.1d. show the same model with a structured meshing scheme. One will notice how nicely aligned the dual sheets are in comparison to the THex model. With DTHex, one would like to get the dual sheets to look more like their mapped-mesh counterparts.

The input to the algorithm is an initial surface triangulation. For biological models, this is the geometric faceted representation that is read in from the file. Geometric features are extracted based on angles between triangles [29]. The first step in the algorithm is to verify valid boundary conditions. This means that every non-boundary edge must be connected to two triangles, and no edges can overlap. It is important to verify these conditions, since many of the biological meshes do not meet these minimum requirements due to translations errors. If errors exist, it is necessary for the user to perform local modification to create a fully conformal mesh.

The second step is to coarsen the mesh. The algorithm used here is an edge-collapsing algorithm, based on node removal. A series of conditions is applied to each edge on the surface, and those edges which meet the conditions are collapsed. Edge swapping is then applied recursively to improve element quality and node valence. Coarsening and edge swapping will be discussed in Chapter 4.

The third step is to smooth the mesh. There are two methods of smoothing that are employed in this research. The first is global smoothing, and the second is local smoothing. These will be discussed in Chapter 5.

The fourth step is to create a tetrahedral mesh on the volume. This may be performed using any tet meshing software[12][13].

The last step is to THex the mesh, and then to further refine by splitting each hex into eight hexes. The final result is a mesh with better quality and decreased node count. Examples will be given in Chapter 7.

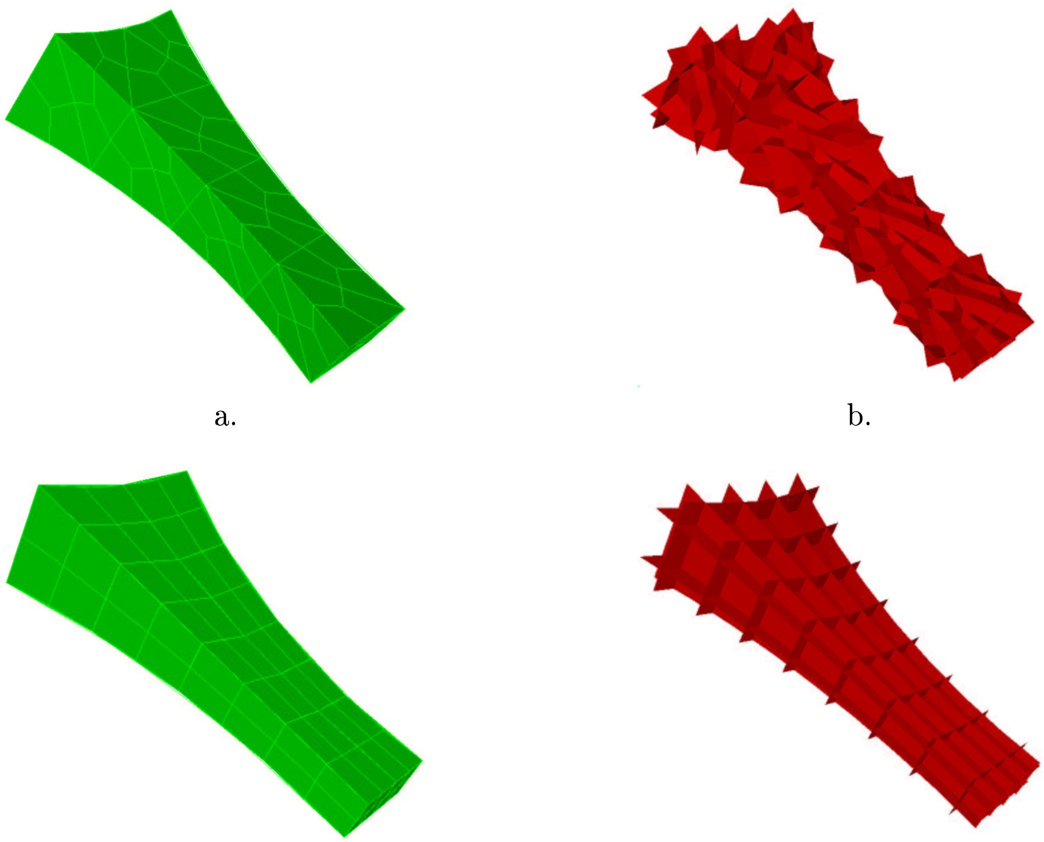


Figure 3.1: Example dual sheets of THex and mapped meshes.

4 COARSENING

This chapter discusses the coarsening algorithm that is used in the DTHex method of hexahedral mesh generation. Coarsening is the most important step of the algorithm, since the surface mesh that is produced after coarsening determines the quality and topology of the final hexahedral mesh. The first section of this chapter discusses the research that has been done on coarsening triangular meshes. The next sections review edge collapsing and edge swapping, the methods used for coarsening in this thesis. The last section presents examples of coarsening.

4.1 Coarsening Literature

Coarsening or decimation is the process of removing entities such as nodes, edges or faces from a triangulation. There are many references on triangular and tetrahedral decimation due to its application in computer graphics[14][15][31][33][37]. The motivation is somewhat different in computer graphics, where a major concern is data reduction, but not necessarily quality improvement. Not as much research has been done in the field of finite element analysis mesh coarsening, where quality is a major concern.

Bank and Xu[1] present a method for coarsening a fine unstructured mesh created through parallel refinement. They use a vertex deletion scheme and define a shape regularity function to denote the quality of the resulting mesh. Borouchaki and Frey[6][10] describe two complimentary means of creating coarse meshes for rigid body analysis using an edge collapsing procedure and an optimization scheme. Finally, Ollivier-Gooch[27] develops a new method of coarsening unstructured meshes based on edge contraction and vertex removal. Quality is ensured by proper selection of nodes for removal. The method presented in this thesis is based on the work of Borouchaki and Frey[10]. Additional features are added to handle the cases of highly concave biological meshes.

4.2 Edge Collapsing

Edge collapsing is the process of systematic vertex removal by edge deletion in a finite element mesh. The most difficult problems encountered in edge collapsing are determining a valid stopping criterion, and preserving quality. Determining a valid stopping criterion involves defining what is "coarse enough" without sacrificing surface definition. The approach used in this thesis was to develop a stopping criteria that is individualized based on surface curvature at each point. The work of Frey and Borouchaki of Inria [6][10][11] was used as a basis for the edge collapsing algorithm presented here. The second challenge is ensuring quality. One method used to improve quality is edge swapping, discussed in this chapter. The second method is smoothing or node moving which is discussed in Chapter 5.

4.2.1 Curvature-Based Sizing Function

A curvature-based discrete sizing function is calculated for the surface using a method proposed by Frey and Borouchaki in [11]. No underlying knowledge of surface curvature is needed to calculate an approximation for curvature. One need only know the normal ν_P and tangent vector τ_P at a point on the surface (Figure 4.3). Frey and Borouchaki define the osculating circle of a point P on curve $\gamma(s)$. This approximation reduces to a second-order polynomial Taylor series approximation for small Δs. The point Q can be approximated by the formula:

$$Q = P + \tau_P \Delta s + \frac{\nu_P}{2\rho_P} \Delta s^2$$

where Q belongs to the plane defined by the vector $\vec{PQ}$ and the normal vector at P, ν_P. One may easily solve for an approximation to ρ_P. The algorithm is given below.

Step 1: Define starting point P;

Step 2: Get all attached edges to P;

Step 3: Set ρ_P = large number

For i = 1 to number of edges

Set Q_i = end node of edge i;

Find $\vec{PQ_i} = \vec{P} - \vec{Q_i}$;

Find ν_P = surface normal at P;

Set $\cos(\theta) = \vec{PQ_i} \bullet \nu_P / \|\vec{PQ_i}\|$

Set $\rho_I = \|\vec{PQ_i}\| / 2\cos(\theta)$

If $\rho_I > \rho_P$

$\rho_P = \rho_I$

end If;

end For;

Step 4: return ρ_P;

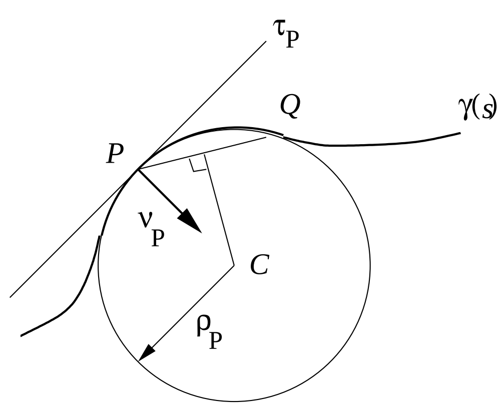


Figure 4.1: Osculating circle approximation

This algorithm loops through the edges surrounding a node, and finds the smallest osculating circle. The ideal edge length can then be determined by defining a coefficient α such that $h(P) = \alpha\rho_P$ where $h(P)$ is the Euclidean length at P (Figure 4.2). A variable ϵ is defined so that

$$\delta/\rho_P \leq \epsilon$$

where δ/ρ_P is the ratio of the distance between the line PQ and the osculating circle and the radius of curvature. This variable ϵ may be referred to as the coarsening factor. It can then be shown that

$$\alpha \leq 2\sqrt{\epsilon(2-\epsilon)}$$

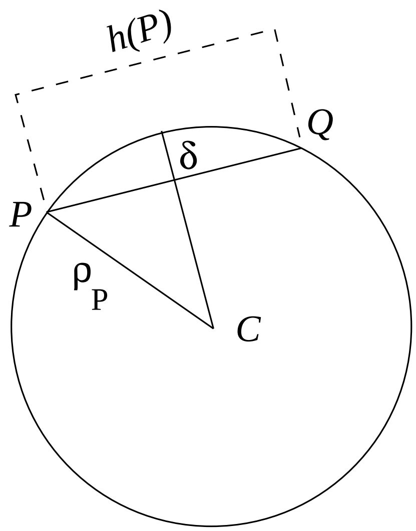


Figure 4.2: Graphical interpretation of distance criterion

4.2.2 Edge Collapse Algorithm

The basic edge collapse algorithm shown below is a modification of [10].

- Let AB be an edge such that $l = L_n(AB)$ and let h_a, h_b be the sizing function parameters at A and B.
- Let $P_0(=A), P_1, \ldots, P_n$ be the vertices adjacent to B given in cyclic order counter-clockwise to the surface normal.
- Compute $l_i = L_n(AP_i), 2 \leq i \leq n-1$.

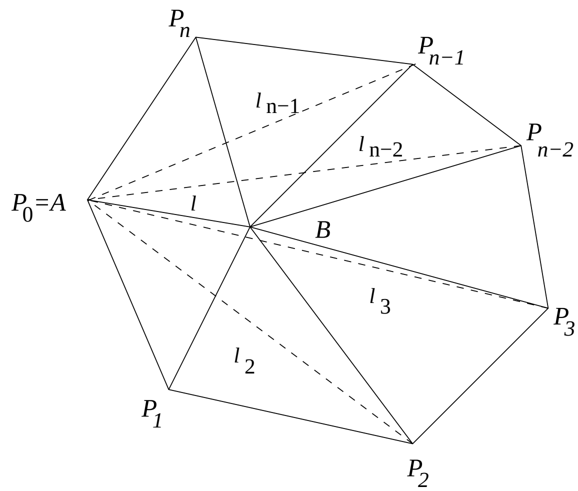


Figure 4.3: Edge Collapsing Procedure

- If $\forall i, 2 \leq i \leq n-1, l_i > l$ and $l_i < \frac{1}{l}$, collapse B to A.

A discrete curvature-based sizing function is defined at each node $h(P)$ as described in the previous section. Given the size at each point, a normalized edge length may be determined using a linear interpolation scheme between points which yields the following formula for the normalized edge length.

$$l(AB) = \frac{d(AB)}{h(B) - h(A)}[Log\frac{h(B)}{h(A)}]$$

The goal of the algorithm is to coarsen such that all edge lengths on the surface have a normalized edge length equal to 1. Methods to improve quality of these meshes will be discussed in the next section. An additional parameter is added to limit the size of any single edge to a user defined maximum, so that in the case of a perfectly flat surface, the user may determine a limit on edge length.

4.3 Edge Swapping

Edge collapsing can severely reduce element quality on the surface mesh. To avoid the disintegration of quality, edge swapping is implemented. Edge swapping is the process of swapping the edge between two triangles to join the opposite nodes. In Figure 4.4 the edge AC can be swapped to produce the triangles $\triangle ABD$ and $\triangle BCD$. The edge swapping used in this section is based on a quality calculation combined with a surface normal check. The goal of edge swapping is to equalize angles in each triangle to produce equilateral triangles, thus improving node valence. Originally, the Delaunay criterion was used for edge swapping, but it disintegrated on curved surfaces, so the quality metric was implemented to avoid these problems.

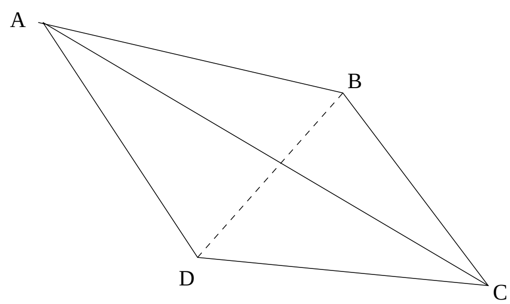


Figure 4.4: An example of edge swapping

- Given two triangles defined by points A, B, C, D where $\triangle ABC$ and $\triangle ACD$ are the vertices of each triangle.
- Define the mean normal for each of the four possible triangles $\triangle ABC$, $\triangle ACD$, $\triangle ABD$, $\triangle CBD$ by averaging the normal vectors for the three nodes.
- For each of the four triangles calculate the quality Q given by the following formula from S.H. Lo[19].

$$Q = \frac{4\sqrt{3}A}{L_1^2 + L_2^2 + L_3^2}$$

- Find the dot product α between the average surface normals for each triangle and the normal of each triangle divided by their respective magnitudes.

- Penalize the quality of each triangle by the formula $Q' = Q\alpha^5$.

- If the minimum quality of the four triangles is improved swap edge AC.

The quality metric from S.H. Lo is modified using a penalization factor raised to the fifth power to account for out-of-plane triangles. Raising alpha to the fifth power is to make sure that triangles close to the tangent plane are not penalized as much as triangles further from the tangent plane.

4.4 Coarsening Examples

This section contains examples that highlight performance of the curvature-based sizing function. The quality metric that is used in this thesis is the shape metric as defined by Knupp (See reference [17]). The average shape quality, minimum, maximum and standard deviation are given for each example. A value of 1.0 indicates a perfectly shaped element (e.g. an equilateral triangle or a cube). A value of 0 indicates an inverted element. The first example is a biological model of a patella. The surface is difficult to coarsen because it has regions that are relatively flat interspersed with very small and steep peaks. A general coarsening algorithm would overlook the small features, but a curvature based coarsening function can capture the small features while coarsening the relatively flat regions. The input to the algorithm is the parameter ϵ. (Recall that ϵ is the coarsening factor defined in Section 4.2.1). A larger value of ϵ yields a coarser mesh.

The next example is a biological model of a tympanic membrane. The model is very thin, which is why there are such small elements around the perimeter. Previous coarsening methods do not adequately capture the sharp curve while maintaining a coarse enough mesh on the rest of the model. The curvature-based coarsening scheme can adequately capture this curvature without sacrificing quality.

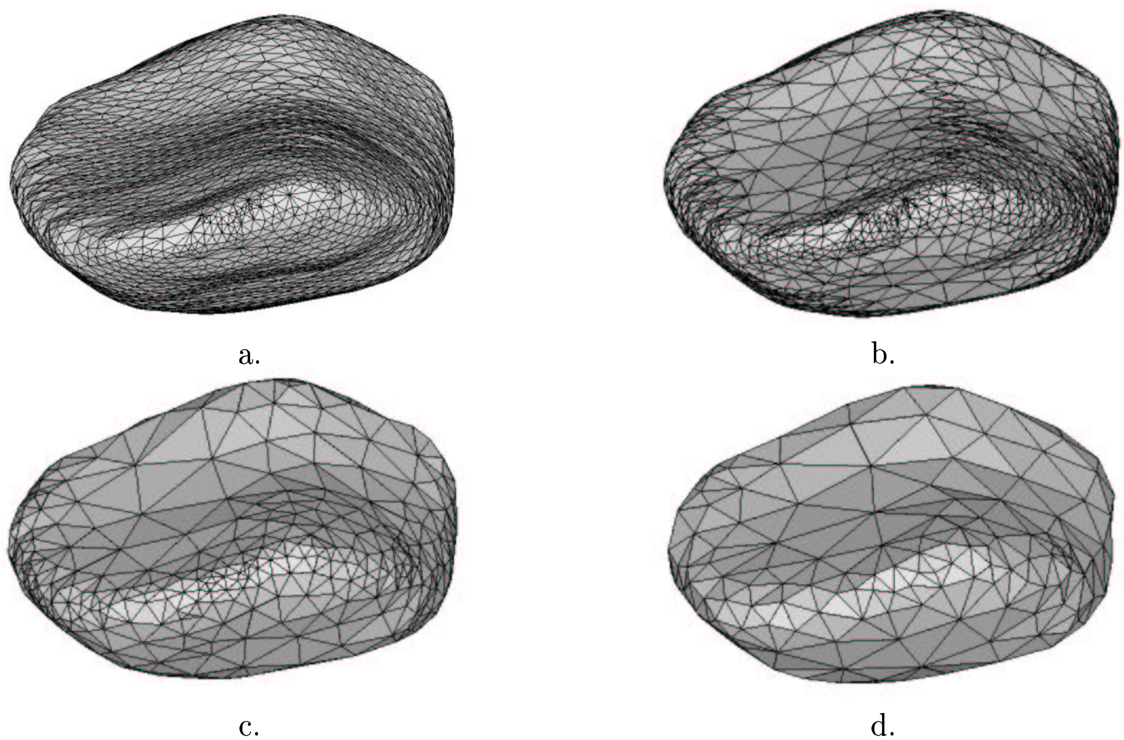


Figure 4.5: Example of coarsening for different ϵ. (a) original mesh (b) $\epsilon = 0.01$ (c) $\epsilon = 0.1$ (d) $\epsilon = 0.5$

Table 4.1: Shape quality for meshes in Figure 4.5.

	Avg.	Std. Dev.	Min.	Max.
Original Mesh	0.6581	0.1712	0.1623	1.000
$\epsilon = 0.01$	0.8311	0.1548	0.2080	1.000
$\epsilon = 0.1$	0.9094	0.07486	0.5972	0.9999
$\epsilon = 0.5$	0.9044	0.07735	0.5447	0.9995

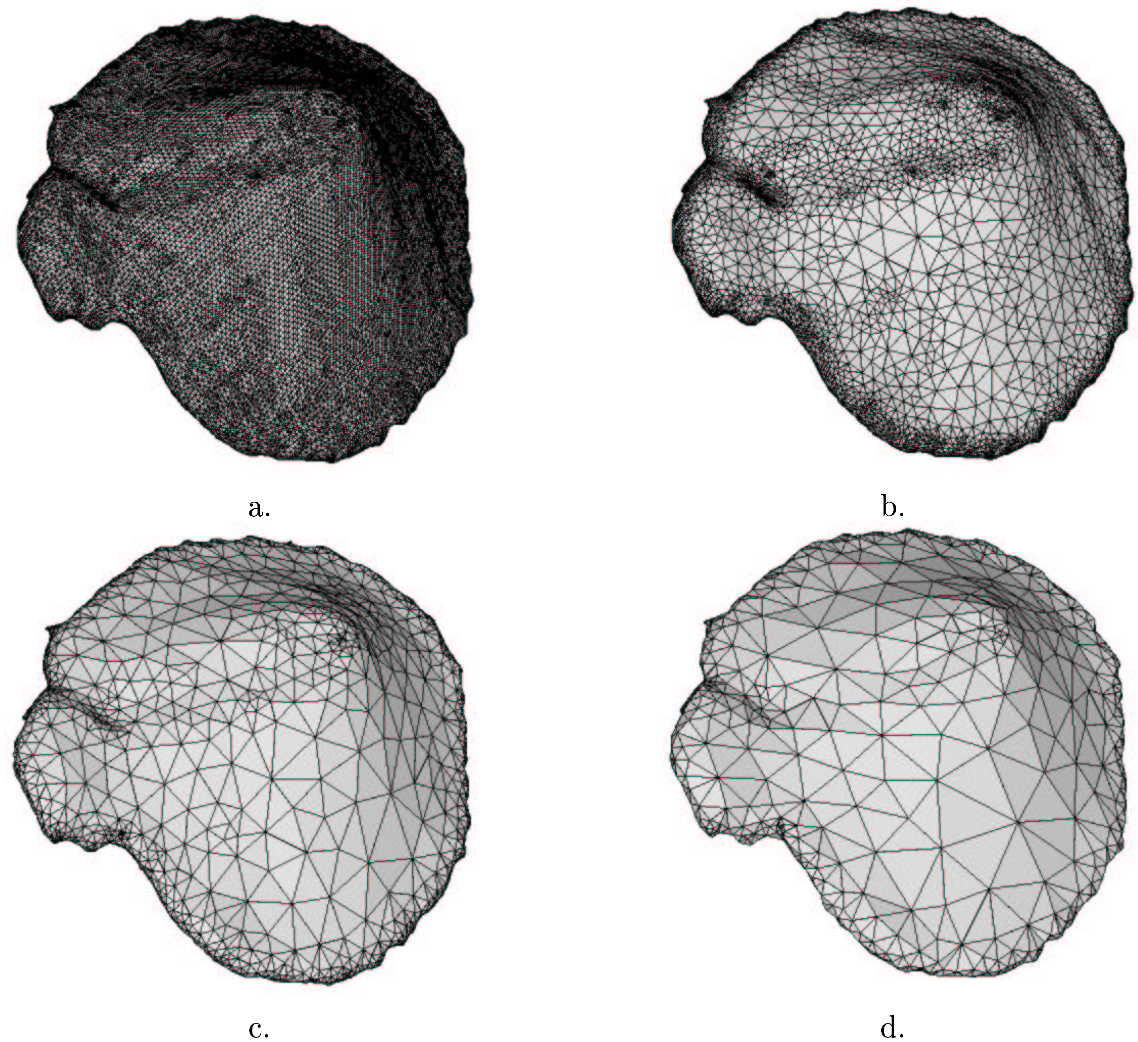


Figure 4.6: Example of coarsening for different ϵ. (a) original mesh (b) $\epsilon = 0.01$ (c) $\epsilon = 0.1$ (d) $\epsilon = 0.5$

Table 4.2: Shape quality for meshes in Figure 4.6.

	Avg.	Std. Dev.	Min.	Max.
Original Mesh	0.9397	0.06240	0.1244	1.000
$\epsilon = 0.01$	0.9105	0.07864	0.3288	1.000
$\epsilon = 0.1$	0.8963	0.09119	0.1933	0.9991
$\epsilon = 0.5$	0.8825	0.1076	0.2715	0.9998

5 SMOOTHING

The major distinction between coarsening meshes for use in finite element analysis, and coarsening meshes for use in computer graphics is the need to maintain good quality elements. One method of improving quality is edge swapping, discussed in the previous chapter. This chapter discusses node moving techniques, known collectively as smoothing procedures. The first section of this chapter focuses on global smoothing using the centroid area pull method. The next section introduces a new smoothing scheme that uses a weighting function to smooth locally. The advantages and disadvantages of each method are compared.

5.1 Global Smoothing

Global smoothing is a process of improving element quality over a surface or volume by iterative node movement. The term global refers to the way the smoothing algorithm is applied over an entire surface, node by node, and does not refer to the solution of a PDE for the surface. During each iteration, node movement should decrease, until it has reached some minimum tolerance. The centroid area pull method [16] is a smoothing method that is applied to nodes in a triangular surface mesh. The goal is to create elements of equal area over the surface. This approach was chosen initially in this research because of the balance between speed and quality. The algorithm given below corresponds to Figure 5.1.

Step 1: Start with a list of nodes on the surface;

Step 2: Loop through the list of nodes;

For i = 1 to number of nodes;

Set $N_i = i^{th}$ node in the list;

For j = 1 to number of triangles attached to N_i;

Calculate total area $A = \sum A_j$;

end For;

For j = 1 to number of triangles attached to N_i;

Calculate $w_j = \frac{A_j}{A}$;

Find C_j = the center of A_j;

Calculate $N\prime_i + = C_j w_j$;

end For;

Set $movement = N_i - N\prime_i$;

If $movement >$ a small tolerance;

Add neighbors of N_i to the list of nodes;

end If;

Step 3: Stop when there are no more nodes in the list;

This method of smoothing produced good quality elements, but in practice took significant time because the entire mesh was smoothed after every coarsening iteration, instead of just smoothing the edges that changed. In addition, it did not respect the sizing functions, but instead tried to make all the mesh edges the same length. For these reasons, it was decided that a local smoothing method was needed to adequately capture mesh features.

5.2 Local Smoothing

Local smoothing is similar to global smoothing in its methodology. However, local smoothing includes only a small number of nodes, instead of the entire surface. This is possible in coarsening because only one vertex is removed at a time. The benefits of this approach are that it significantly increases the speed of the algorithm and that it respects sizing constraints. Like the centroid area pull method, the localized smoothing technique that was employed in DTHex tries to equalize element area. It also averages the sizing functions of each node on the triangle as a secondary

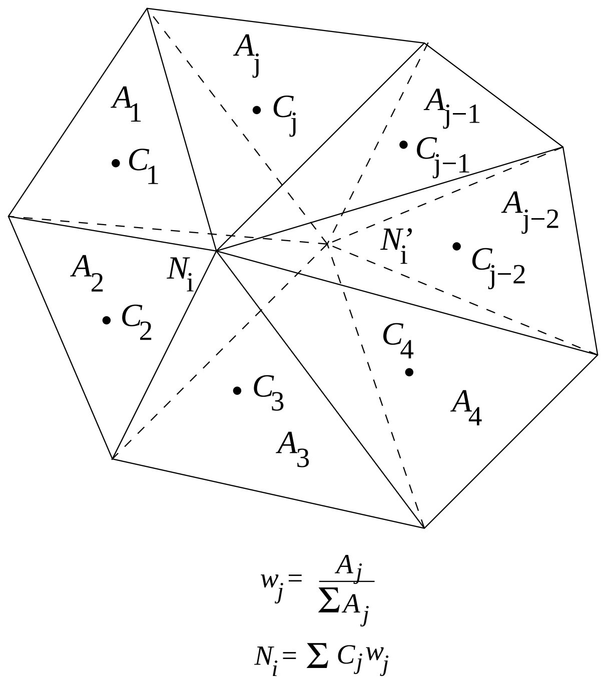


Figure 5.1: Global smoothing

weight. The new node position is a combination of these two weighting functions. Two coefficients c_1 and c_2 are defined, where $c_1 + c_2 = 1$. These factors represent how much the element area and sizing function parameters affect the node movement. Increasing the weight on the triangle size c_1 generally improves the quality, while increasing the weight on the sizing function c_2 more accurately captures the geometry. The algorithm for local smoothing is shown below. See also Figure 5.2.

Step 1: Input a single node N;

Step 2: Input c_1 and c_2;

Step 3: For i = 1 to number of triangles attached to N;

Calculate total area $A = \sum A_i$;

Calculate sum of sizing functions $S = \sum 1/S_j$;

end For;

For i= 1 to number of triangles attached to N;

Calculate $w1_i = \frac{A_i}{A}$;

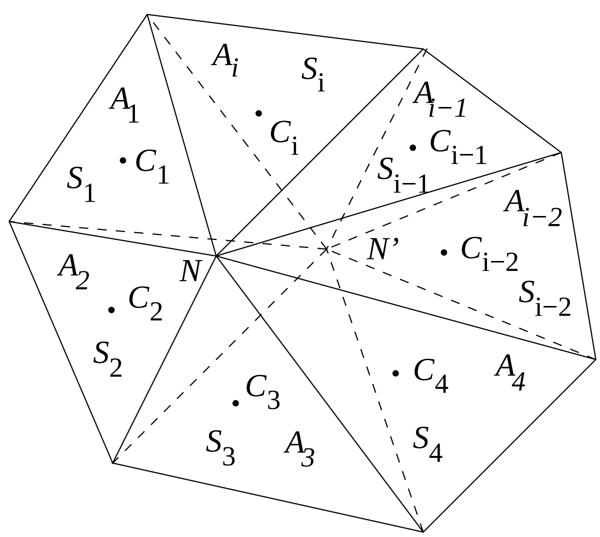
Calculate $w2_i = \frac{1/S_i}{S}$;

Find $C_i =$ the center of A_i;

Calculate $N\prime + = C_i(c_1 w1_i + c_2 w2_i)$;

end For;

Localized smoothing is applied to the node to which each edge is collapsed. It is also applied to nodes after edge swapping. Local smoothing does not optimize quality as well as a global technique, because each node is only moved once. However, the benefits of increased speed and sizing accuracy outweigh the costs. In the examples, the local smoothing actual outperformed global smoothing in quality measurements as well due to the fact that global smoothing would reach a maximum number of iterations before it converged.



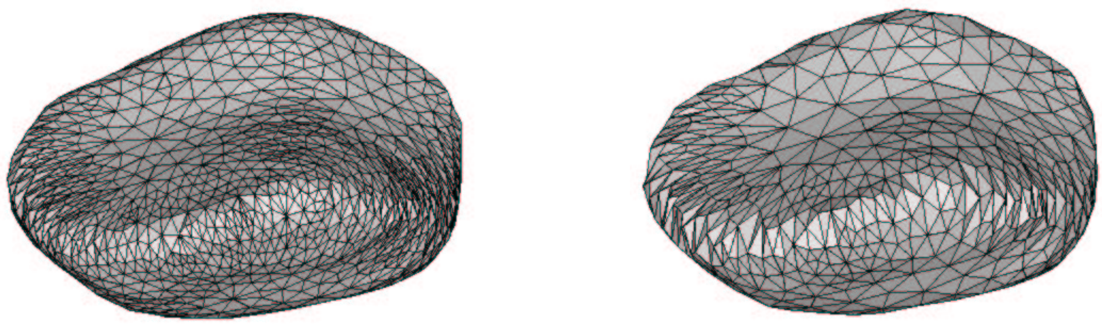
$$w1_i = \frac{A_i}{\sum A_i} \qquad w2_i = \frac{1/S_i}{\sum 1/S_i}$$

$$N' = \sum C_i (c_1 w1_i + c_2 w2_i)$$

Figure 5.2: Local smoothing

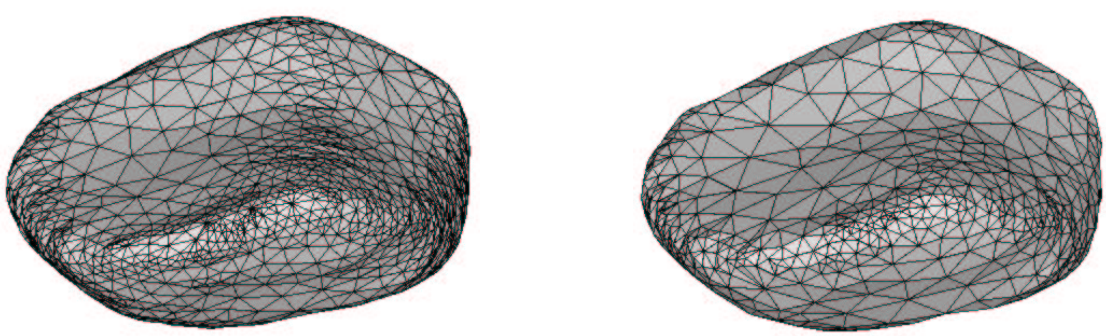
5.3 Smoothing Examples

We return to the biological model to compare different smoothing techniques. In this example, quality and speed were compared for two different values of ϵ with global and local smoothing. The global smoothing technique took 566 seconds for ϵ=0.01 versus 1.19 seconds for local smoothing on ϵ=0.01. Additionally, the local smoothing actually outperformed global smoothing in quality comparisons for both ϵ.



a. b.

Figure 5.3: Global smoothing with (a) $\epsilon = 0.01$ and (b) $\epsilon = 0.05$



a. b.

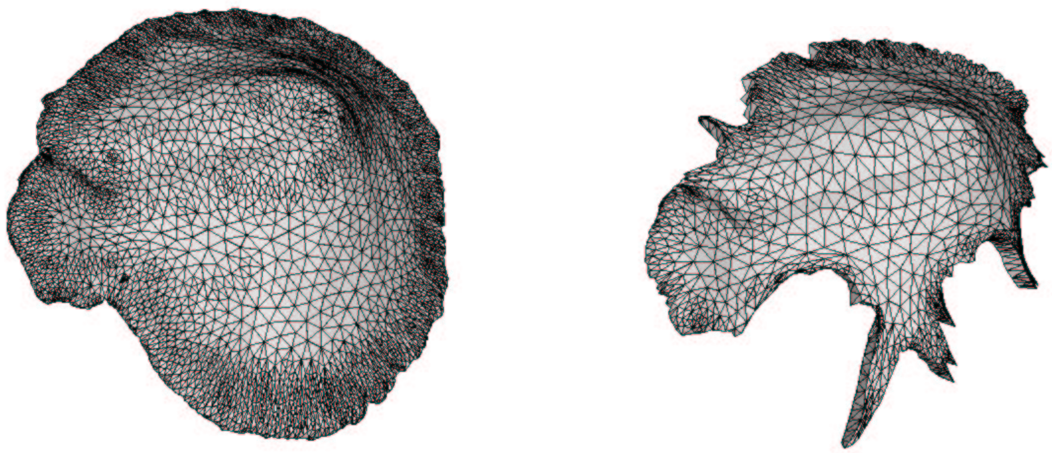
Figure 5.4: Local smoothing with (a) $\epsilon = 0.01$ and (b) $\epsilon = 0.05$

In the second biological model of the tympanic membrane the global smoothing method caused the mesh to collapse away from the boundary. With quality and speed improved, it was found that local smoothing was the only viable option for coarsening. It is interesting to point out that the direction of time increase is reversed between global smoothing and local smoothing. In other words, the increase in ϵ results in a decrease in time for global smoothing, but an increase in time for local smoothing. This is because in global smoothing the time to smooth is proportional to the number of elements in the mesh, while in local smoothing, the time to smooth is proportional to the number of edge collapses. Figures 5.5 and 5.6 show that the local smoothing has a greater differential between areas of low and high curvature than the global smoothing.

Table 5.1: Comparison of smoothing techniques for Figures 5.3 and 5.4.

	ϵ	Shape	CPU Sec.
Global Smoothing	0.01	0.8073	566
	0.05	0.7926	26.3
Local Smoothing	0.01	0.8311	1.19
	0.05	0.8954	1.97

The local smoothing did not outperform the global smoothing on very simple models such as a sphere. However, for the majority of applications it is believed that local smoothing is a significant improvement over global smoothing for coarsening applications.

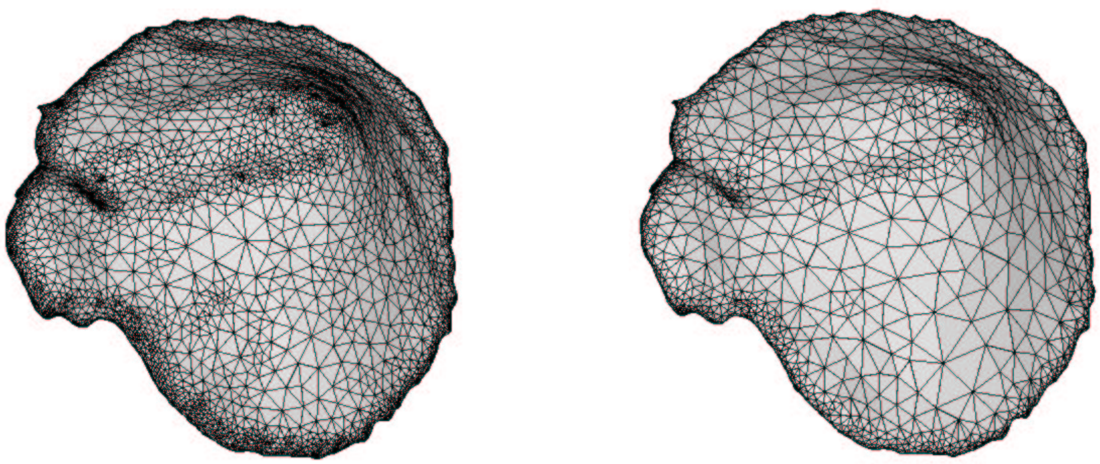


a. b.

Figure 5.5: Global smoothing with (a) $\epsilon = 0.01$ and (b) $\epsilon = 0.05$

Table 5.2: Comparison of smoothing techniques for Figures 5.5 and 5.6.

	ϵ	Shape	CPU Sec.
Global Smoothing	0.01	0.7662	631
	0.05	0.6587	455
Local Smoothing	0.01	0.9105	16.6
	0.05	0.8987	21.2



a. b.

Figure 5.6: Local smoothing with (a) $\epsilon = 0.01$ and (b) $\epsilon = 0.05$

6 DTHEX

DTHex is the process of converting a triangular surface mesh into a valid hexahedral mesh using the coarsening and smoothing tools discussed in Chapters 4 and 5. The last three steps of the DTHex procedure are tetrahedral meshing, THexing and dicing. The tetrahedral meshing is completed using procedures described by George[12][13] and included in the CUBIT software. THexing and Dicing are discussed in this chapter.

6.1 THex

THexing was introduced in Chapter 1 as the process of splitting every tetrahedron in the mesh into four hexahedra. THexing is a simple procedure, and it is used frequently because of its ease of implementation. For example, Pelessone and Charman [30] use this procedure in adaptive non-linear structural analysis where local refinement of a hexahedral mesh is needed. The advantages of THexing as introduced in Chapter 2 are its speed, lack of user intervention, geometric generality, orientation insensitivity, and element size control. The disadvantages are that it has poor boundary sensitivity, increased element count, and poor element quality. Another problem with the THexing is the node projection problem. When edges that are on convex surfaces are split during THexing, the interior node must be projected inward. See Figure 6.1. This often creates inverted elements. There are different approaches to solving this problem, but those are outside the scope of this research. The assumption here is that the curvature-based coarsening will help to alleviate that problem by using smaller elements on highly curved segments.

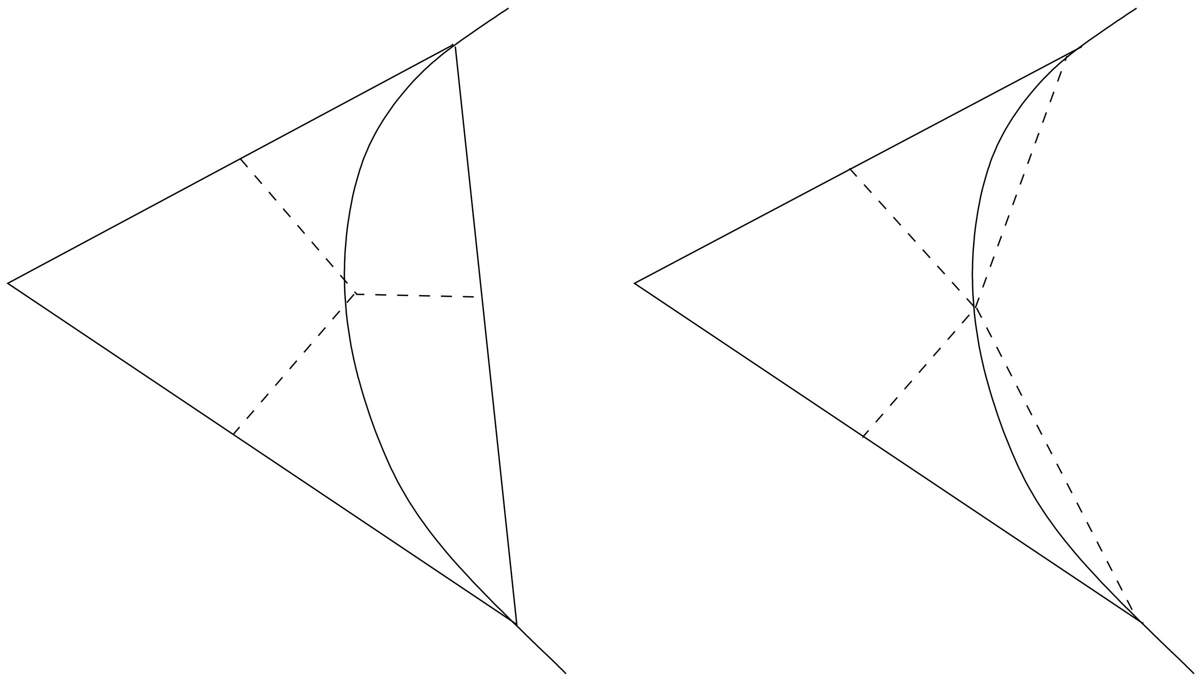


Figure 6.1: Node projection problem on concave surfaces.

6.2 Dicing

Dicing is a refinement procedure described by Melander [22] as a method of creating multi-million element meshes. Using the dicing method, every hexahedral element can be subdivided into smaller hexahedra. This is an 'h' modification that can improve analysis accuracy by producing a mesh with smaller element size, and improved angles. Dicing should be used with caution on mesh with a large number of nodes, since the increase in node count may exceed analysis capabilities. However, if used in conjunction with proper coarsening techniques, it can help to increase the overall quality of the mesh. The next chapter shows the dicing procedure in each of the DTHex examples.

7 DTHEX EXAMPLES

The main application of this research is for a specific class of hexahedral mesh generation, namely biological models. The stated goal was to improve element quality and reduce node count for these complex meshes. With that purpose in mind, this chapter highlights the features of the DTHex algorithm with several example problems. The chapter begins with fairly trivial problems to highlight some of the features, then extends to more complex biological problems. All of the examples will be compared to the existing THex approach in terms of quality and node count.

7.1 Sphere

The first example is a simple sphere. The sphere is an ideal mesh model because it is entirely convex. Because it is convex, the mesh does not develop any negative Jacobian elements when it is THexed as all projections are outward. The original sphere is meshed with a triangle scheme at an edge length size of one tenth the radius. The surface is coarsened with a coarsening factor of $\epsilon = 0.1$. (Recall that ϵ refers to the ratio of the distance from the edge to the surface divided by the radius of curvature from Chapter 4). The radius of curvature for a sphere is equivalent to the radius of the sphere. If the radius of the sphere is equal to 1, the distance from the edge to the sphere is no greater than 0.1 at any point on the sphere. This is indeed the case for this coarsening example. Local smoothing was used in this example to create a mesh of good quality, whereas global smoothing caused the entire mesh to converge to a single point. The final mesh has an average shape quality of 0.8298 compared to a value of 0.6018 for the THexed mesh (see Table 7.1). Figure 7.1 shows the model at various stages in the DTHex process. Figure 7.2 compares the THex approach to the DTHex approach. The final number of nodes with THexing is over 18000 while with DTHex it is around 2500. Figures 7.3 and 7.4 show histograms of element quality over the volume.

Table 7.1: Shape quality for meshes in Figure 7.2.

	Avg.	Std. Dev.	Min.	Max.
THex Mesh	0.6018	0.08286	0.3439	0.8230
DTHex Mesh $\epsilon = 0.1$	0.8298	0.08310	0.5332	0.9809

7.2 Patella

The second example is a patella model (See Figure 7.5). The model appears to be very smooth, but it has regions of sharp dips and valleys with high curvature. The non-curvature-based coarsening function did not perform well on this analysis because it did not capture these features. The curvature-based coarsening algorithm was able to maintain the relative size of features while creating as coarse a mesh as possible in the regions of low curvature. The final hexahedral mesh did have some negative Jacobian elements, but there were many fewer than with the non-curvature-based coarsening function. Further research will be necessary to determine if it is possible to entirely eliminate these negative Jacobian elements. One temporary solution is to constrain the nodes so they don't project to the surface during the THexing process. Since the mesh approximates the surface curvature by its sizing function this may be a valid option if the user does not mind sacrificing some of the surface definition. The user would be able to specify the amount of desired surface definition by setting the ϵ variable. Figure 7.6 compares the THex approach to the DTHex approach. The final number of nodes with THexing is 162000 while with DTHex it is around 120000.

Figures 7.7 and 7.8 show histograms of element quality over the volume. The average quality shown in Table 7.2 is 0.5424 for the THex mesh and 0.6618 for the DTHex mesh.

Table 7.2: Shape quality for meshes in Figure 7.6.

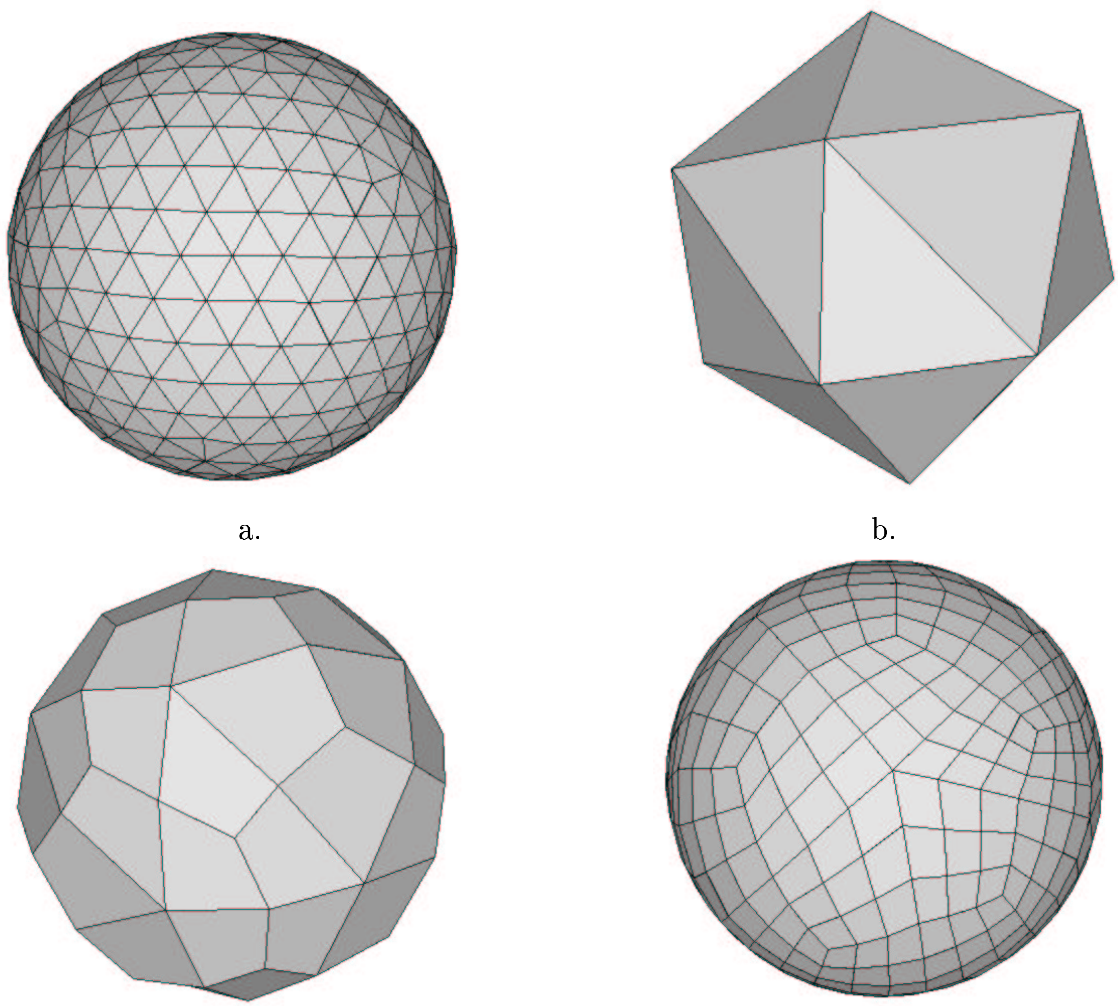
	Avg.	Std. Dev.	Min.	Max.
THex Mesh	0.5424	0.1007	0.000	0.8084
DTHex Mesh $\epsilon = 0.1$	0.6618	0.1114	0.000	0.9549

7.3 Tympanic Membrane

The third example is a tympanic membrane. The tympanic membrane is part of the inner ear. This mesh is difficult because it is quite thin. This makes coarsening around the edges of the membrane difficult where there is a 180 degree turn over a very short distance. Curvature-based coarsening performed as expected by maintaining relatively small elements on the outer border, while allowing larger elements on the flat regions. Another problem with this model is the number of elements in the initial mesh. Re-meshing is not possible with faceted biological models because of the way they are defined. Thus the coarsest possible hexahedral mesh for this model is over 1.7 million nodes before the introduction of the DTHex scheme. DTHex decreases that number to 300000 nodes while also improving the quality and respecting curvature. Figure 7.9 shows the mesh after each step of DTHex. The THex approach is shown in Figure 7.10. Quality histograms are shown in Figures 7.11 and 7.12. Average quality shown in Table 7.3 is 0.6320 for the DTHex approach compared to 0.5807 for the THex mesh.

Table 7.3: Shape quality for meshes in Figure 7.10.

	Avg.	Std. Dev.	Min.	Max.
THex Mesh	0.5807	0.07854	0.000	0.8190
DTHex Mesh $\epsilon = 0.1$	0.6320	0.1300	0.000	0.9562



a. b.

c. d.

Figure 7.1: A simple sphere meshed using the DTHex scheme with $\epsilon = 0.1$ (a) Original mesh (728 triangles, 1092 edges, 366 nodes) (b) Mesh after coarsening (20 triangles, 30 edges 12 nodes) (c) Mesh after THexing (80 hexes, 60 faces, 120 edges, 125 nodes) (d) Final hexahedral mesh (2160 hexes, 540 faces, 1080 edges, 2473 nodes)

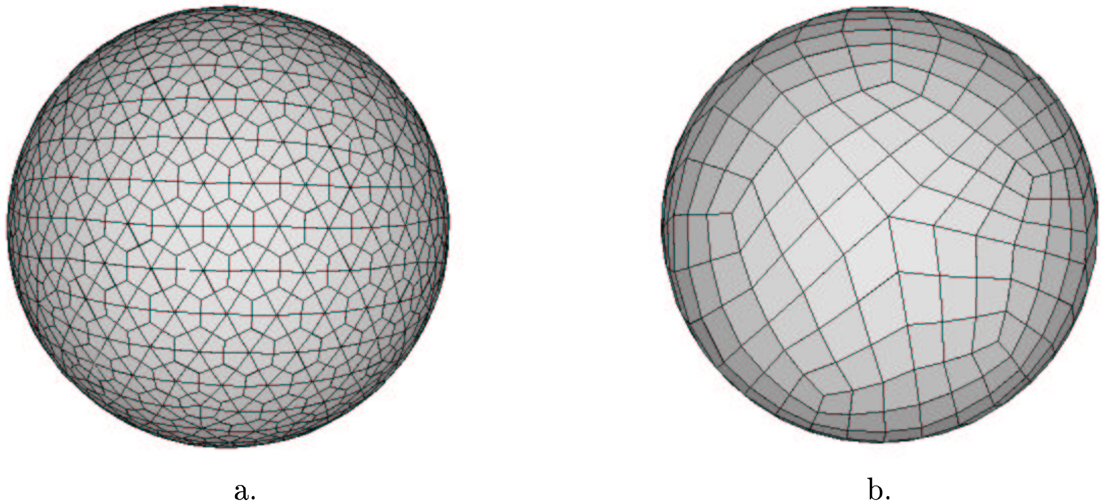


Figure 7.2: A simple sphere meshed with the THex and DTHex schemes (a) THex mesh (16176 hexes, 2184 faces, 4368 edges, 18647 nodes) (b) DTHex mesh (2160 hexes, 540 faces, 1080 edges, 2473 nodes)

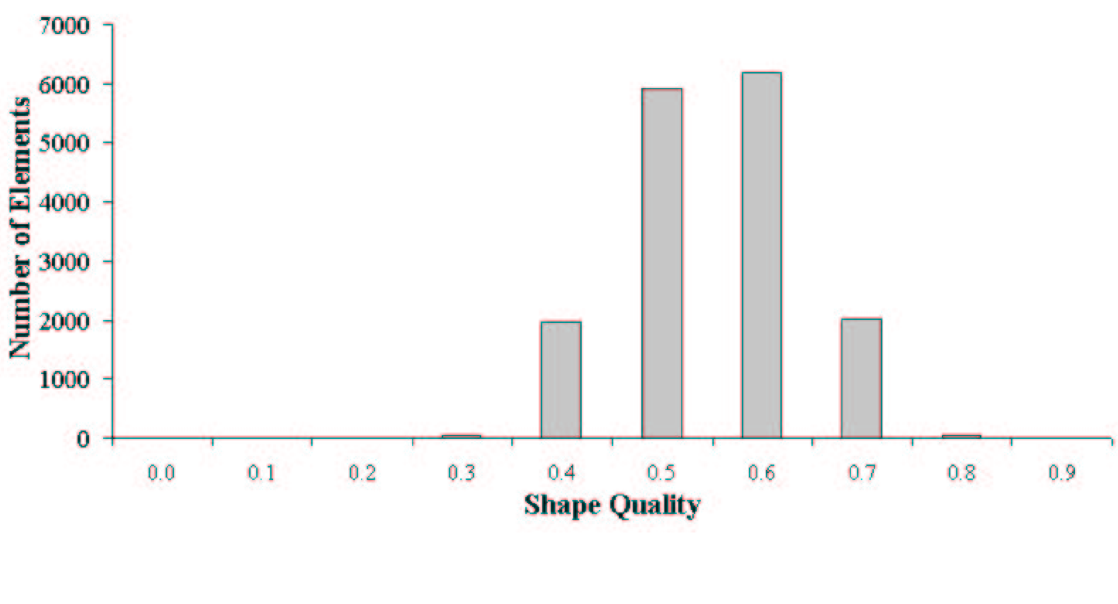


Figure 7.3: Quality graph for sphere meshed with THex.

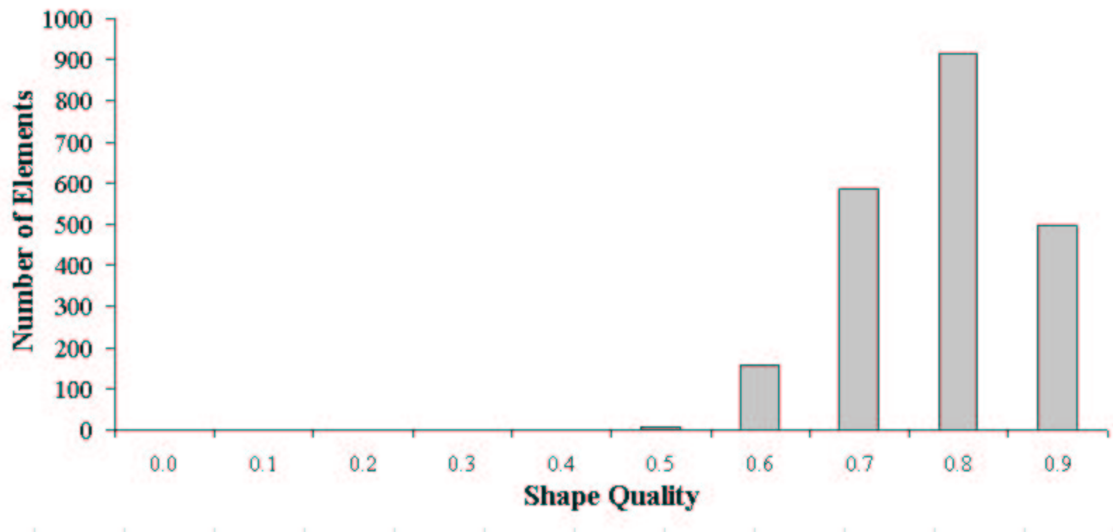


Figure 7.4: Quality graph for sphere meshed with DTHex.

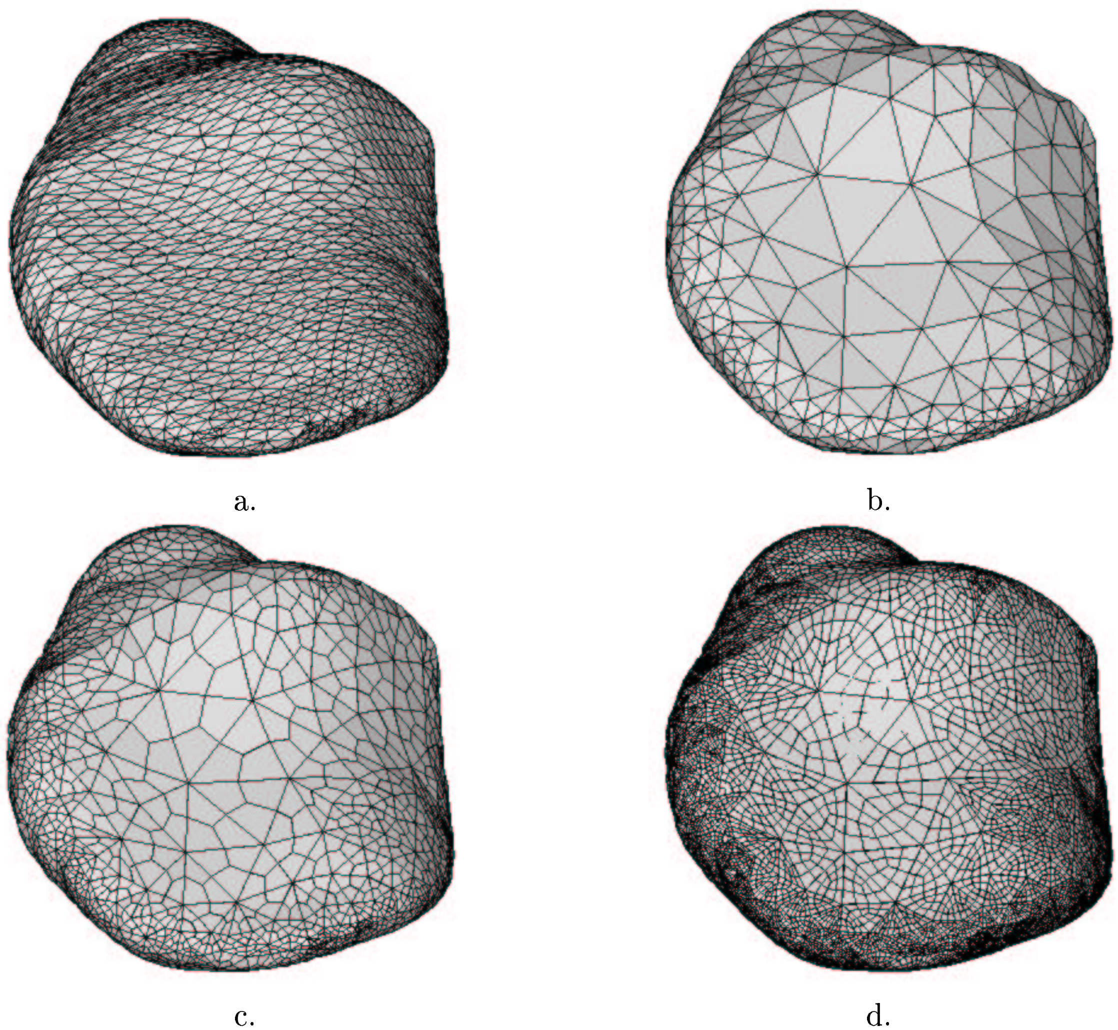


Figure 7.5: A patella meshed using the DTHex scheme with $\epsilon = 0.1$ (a) Original mesh (5862 triangles, 8793 edges, 2933 nodes) (b) Mesh after coarsening (1146 triangles, 1719 edges, 575 nodes) (c) Mesh after THexing (13704 hexes, 3438 faces, 6876 edges, 16647 nodes) (d) Final hexahedral mesh (109632 hexes, 13752 faces, 27504 edges, 118955 nodes)

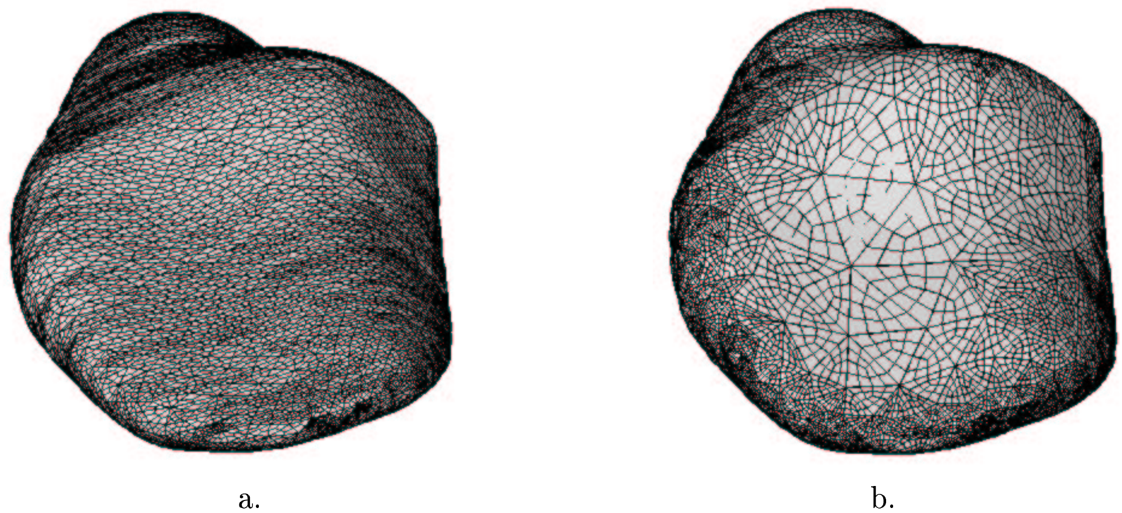


Figure 7.6: A patella meshed with the THex and DTHex schemes (a) THex mesh (141156 hexes, 17586 faces, 35172 edges, 161731 nodes) (b) DTHex mesh (109632 hexes, 13752 faces, 27504 edges, 118955 nodes)

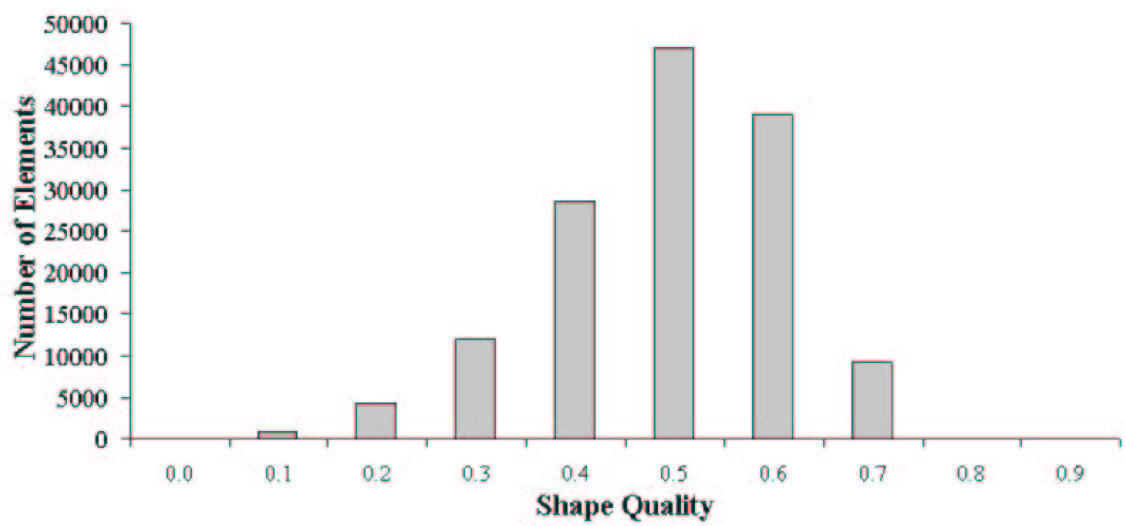


Figure 7.7: Quality graph for patella meshed with THex.

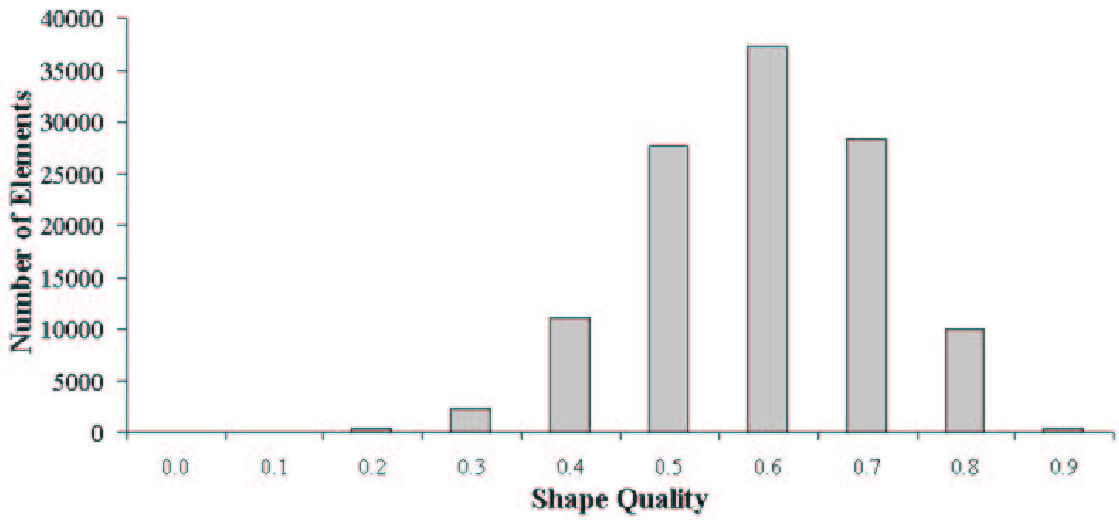


Figure 7.8: Quality graph for patella meshed with DTHex.

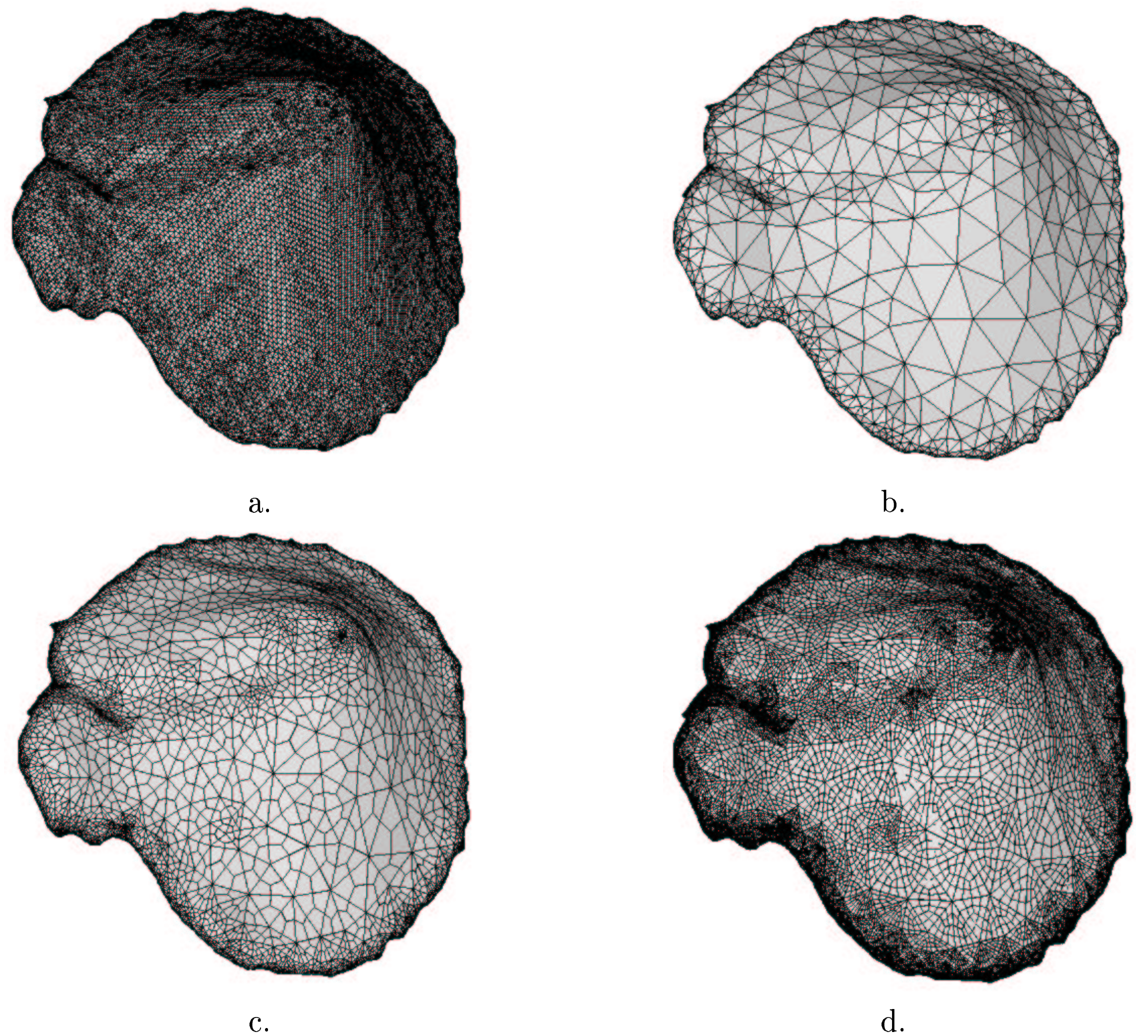


Figure 7.9: A tympanic membrane meshed using the DTHex scheme with $\epsilon = 0.1$ (a) Original mesh (49340 triangles, 74010 edges, 24672 nodes) (b) Mesh after coarsening (4424 triangles, 6636 edges, 2214 nodes) (c) Mesh after THexing (33144 hexes, 13272 faces, 26544 edges, 42899 nodes) (d) Final hexahedral mesh (265152 hexes, 53088 faces, 106176 edges, 297933 nodes)

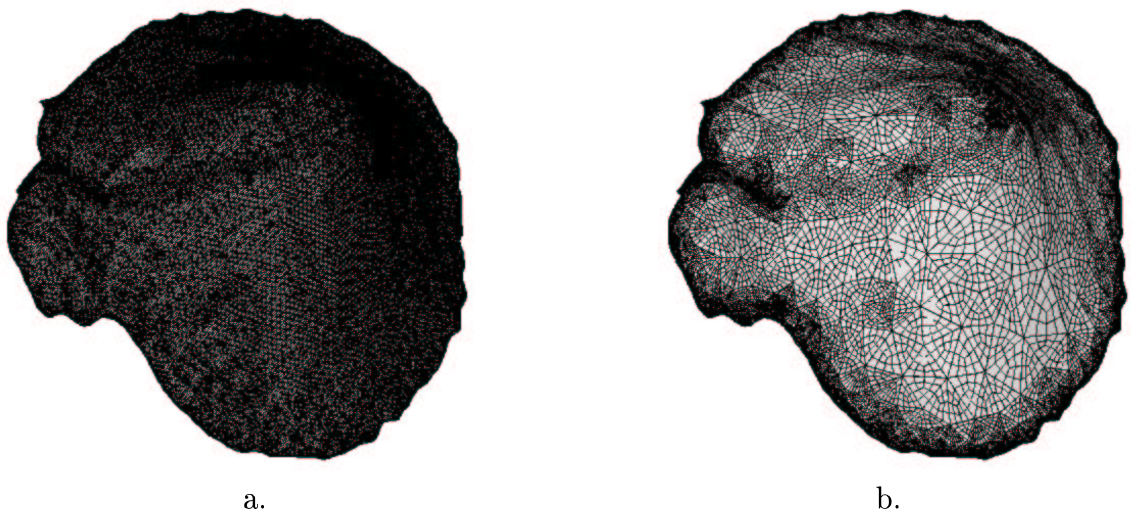


Figure 7.10: A tympanic membrane meshed with the THex and DTHex schemes (a) THex mesh (1546064 hexes, 148020 faces, 296040 edges, 1748575 nodes) (b) DTHex mesh (265152 hexes, 53088 faces, 106176 edges, 297933 nodes)

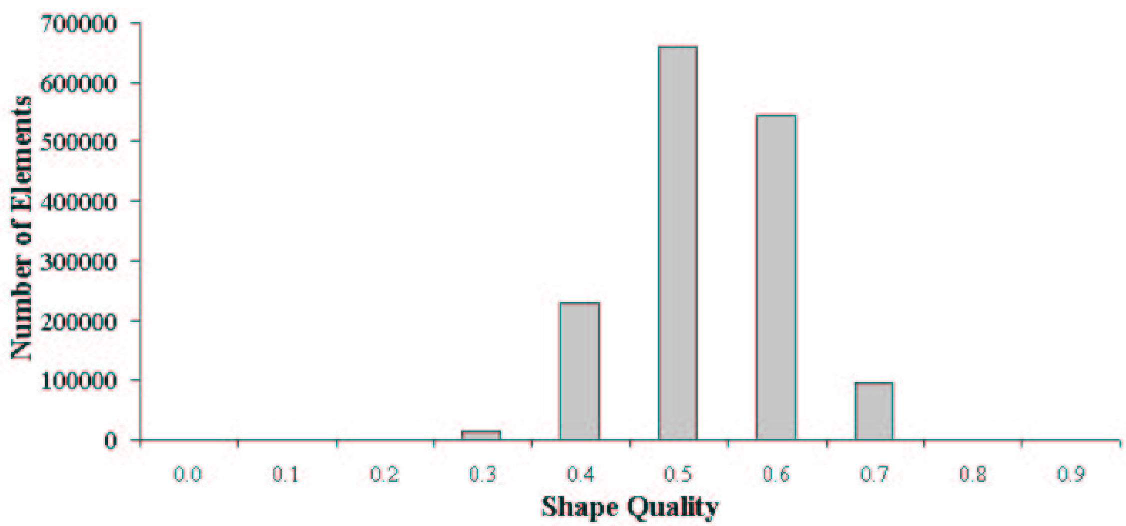


Figure 7.11: Quality graph for tympanic meshed with THex.

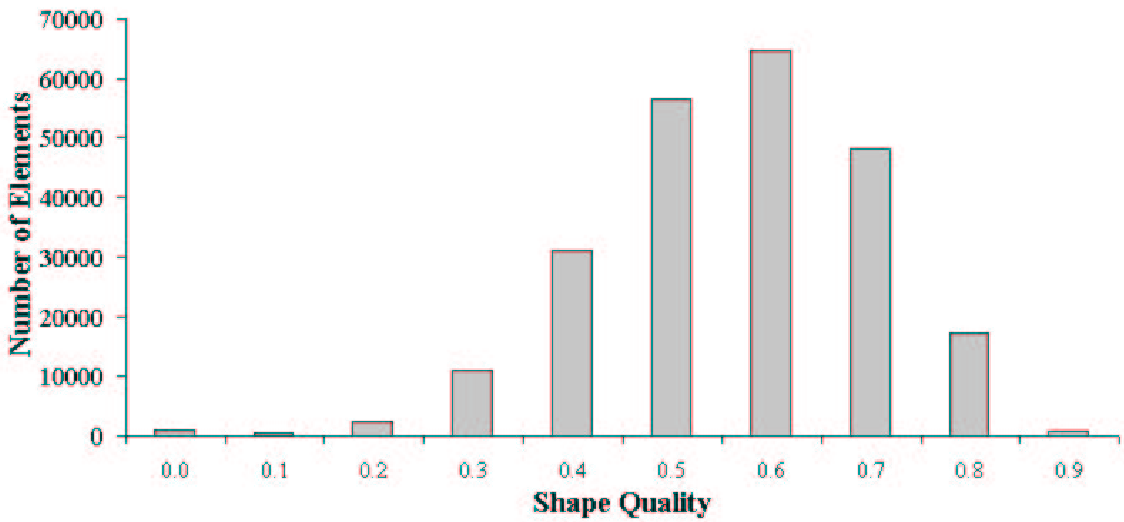


Figure 7.12: Quality graph for tympanic meshed with DTHex.

8 CONCLUSION

Hexahedral meshing is challenging because of the layered nature of hexahedral sheets. For this reason, it has been difficult to come up with an all-hexahedral meshing scheme that is robust enough to produce good meshes on all geometries. THexing is one approach that has been used with moderate success. The work presented in this thesis has shown that the DTHex method of producing all-hexahedral meshes can produce meshes with better quality and fewer elements than the classical THex approach. An additional benefit of the method is the ability to create meshes that approximate surface curvature.

This research has potential for great impact in the field of biological meshing. While there are other coarsening algorithms available in other meshing packages, the synthesis of parallel coarsening and refinement with curvature-based sizing functions, and local smoothing is unique to this research. The improved meshes will allow biological analysis to become more viable by decreasing the computing power necessary to perform the analyses.

This research also provides a framework for further studies on improving node projection techniques. It may be possible to completely eliminate the inverted elements that are formed when nodes on concave surfaces are projected inward. Further studies may also be done on automatically determining which ϵ to use during coarsening.

9 GLOSSARY

The work of this thesis was accomplished using the meshing software package CUBIT. CUBIT is released by Sandia National Laboratories. Since much of the vocabulary used throughout this text is specific to the CUBIT software package, definitions will be provided in this section.

9.1 Geometric Entities

- A *vertex* refers to a zero-dimensional point on the geometric model. Each vertex is represented by a 3-dimensional coordinate in Cartesian space.

- A *curve* is a line or arc that connects vertices. An open curve contains two vertices–one on each end. A closed curve connects to itself, and contains only one vertex. An example of a closed curve is the top of a cylinder.

- A *surface* is defined by a collection of curves or vertices. Closed surfaces may not contain any curves, such as a sphere. A toroidal surface is a surface with a hole. A donut is a 1-toroidal surface, meaning it has one hole. Surfaces may be two or three dimensional.

- A *volume* is a collection of surfaces that form a closed loop in space. In this context, a volume is always three-dimensional.

9.2 Mesh Entities

- An *element* refers to a three-dimensional mesh finite element. An element may be a hexahedron, tetrahedron, pyramid, or one of many more unusual shapes. In this thesis, the term element will apply to hexahedra and tetrahedra only.

- A *node* is a zero-dimensional entity that defines the corners of an element. Each vertex will also have a corresponding node, and each curve may have one or more nodes attached to it. A node has a geometric position in Cartesian space, and also contains information about the elements and faces that it is attached to. An element may also have mid-side nodes in the case of higher-order approximations. These higher order approximations are referred to by the number of nodes per element. For example, a hexahedral element with 20 nodes would be referred to as a Hex20 element.

- The term *edge* refers to the one-dimensional edges of a finite element. Each edge may be connected to one or more finite elements. An edge will always connect two nodes, and it is always straight.

- The term *face* may be used for a quadrilateral mesh face if it is composed of four nodes, or a triangle mesh face if it has three nodes. To avoid confusion, the term face will be reserved for quadrilateral faces, and the term triangle will be used for three-sided faces. If three of the nodes on a quadrilateral face become collinear or concave with respect to the center of the face, the face is given the term "inverted" or "negative Jacobian". The quality of an inverted face is zero.

- A *hexahedron* is a six-sided finite element resembling a brick. Each hexahedron is composed of eight zero-dimensional nodes, twelve one-dimensional edges, and six two-dimensional faces. A hexahedron may also become inverted when three points become collinear or concave.

9.3 Dual Entities

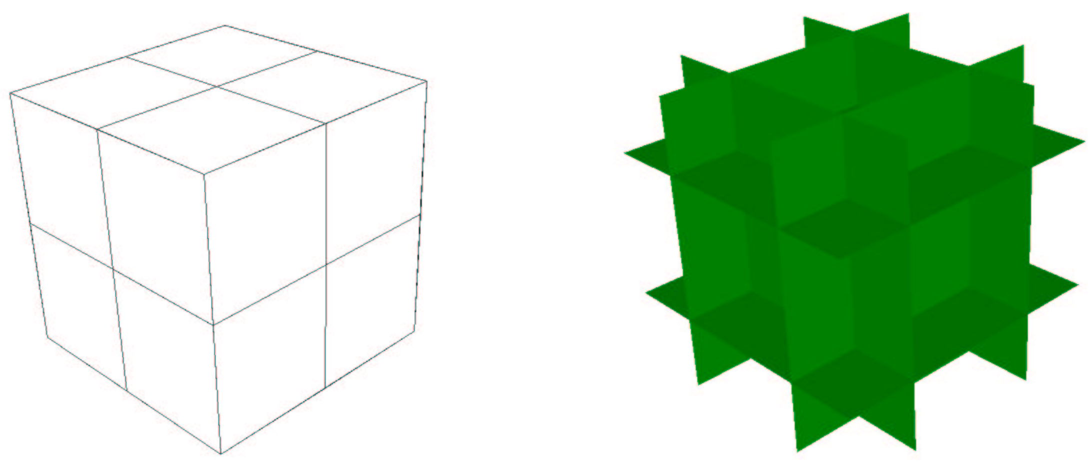
The dual of the mesh is a useful tool proposed by Murdoch[26] to represent the connectivty of a hexahedral mesh. Hexahedral meshes are distinctive in that they form layers of elements. The dual is a graphical way to visualize these layers. Since the dual can be thought of as the inverse of the mesh, each entity has a mesh counterpart. Figure 9.1 shows a mesh and its corresponding dual.

- An *centroid* is the point where three dual chords intersect. The location of a centorid may be found by connecting opposing faces with a straight line through their center points. A centroid is the dual equivalent of a mesh node. A centroid is a zero-dimensional member.

- A *chord* is a one-dimensional member that connects the centroids of the dual. In two-dimensional space, the dual chord can be found by connecting opposing edges of a quadrilateral face through the midpoints on each edge with a straight line. A chord is the dual equivalent of an edge.

- A $2-cell$ is a two-dimensional member that is created when four dual chords intersect to form a quadrilateral. A 2-cell is the dual equivalent of a face.

- A $3-cell$ is a three-dimensional member that is created when six 2-cells intersect to form a hexahedron. The complete set of 3-cells that is created by connecting all of the midpoints on opposing faces of hexahedra is known as the dual of the mesh. A 3-cell is the dual equivalent of a mesh element.



a. b.

Figure 9.1: A mesh (a.) and its dual (b.)

Table 9.1: Relationship between Geometry, Mesh, and Dual Entities.

Dimension	Geometry	Mesh	Dual
0	Vertex	Node	Centroid
1	Curve	Edge	Chord
2	Surface	Face	2-cell
3	Volume	Element	3-cell

9.4 Other Meshing Terms

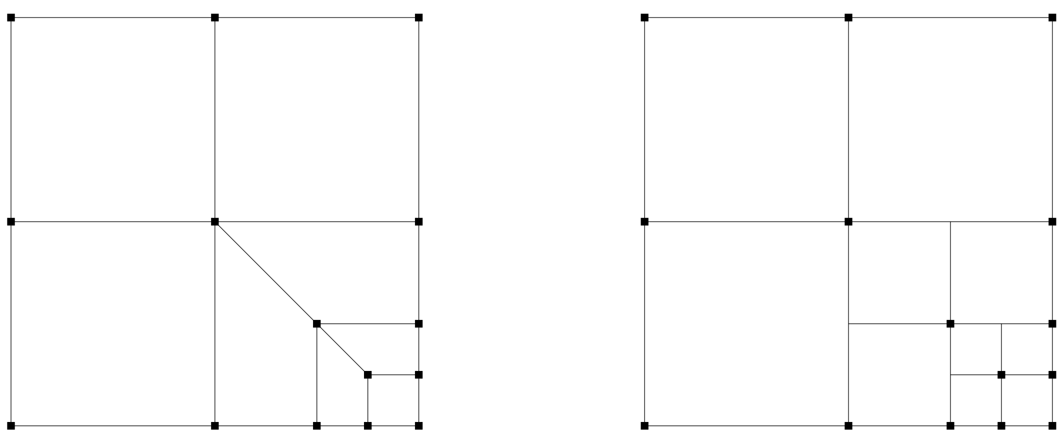
- A *corner* node is located at the intersection of three geometric curves in three dimensions, or two geometric curves in two dimensions.
- An *edge* node is located on a curve, but not at the endpoint.
- A *side* node is located on the interior of a surface in three dimensions. There are no sides nodes in two dimensions.
- An *interior* node is located on the interior of a surface in two dimensions, or the interior of a volume in three dimensions.
- The *valence* of a node is the number of mesh edges attached to that node.
- A mesh is $manifold$ if every edge in the mesh connects exactly two nodes (see Figure 9.2. In other words, elements will only intersect other elements at their nodes. A manifold mesh is generally required for any analysis, and will be assumed for all models in this thesis.
- A quadrilateral mesh is *structured* if every interior node has a valence of four, every side node has a valence of three, and every corner node has a valence of

two. Mapped and Sub-mapped meshing schemes are both structured meshing schemes.

- A hexahedral mesh is *structured* if every interior node has a valence of six, every side node has a valence of five, every edge node has a valence of four, and every corner node has a valence of three. Figure 9.3 shows a structured mesh created with the submap meshing scheme.

- A hexahedral or quadrilateral mesh is *unstructured* if it has irregular node valence. Most of the meshes in this thesis are unstructured, since they are derived from tetrahedral meshes. Figure 9.4 shows an example of an unstructured mesh.

- A hexahedral mesh is $semi - structured$ if it meets the requirements for a structured mesh in one direction, known as the sweeping direction. See Figure 9.5



a.   b.

Figure 9.2: (a) Example of a manifold mesh and (b) non-manifold mesh

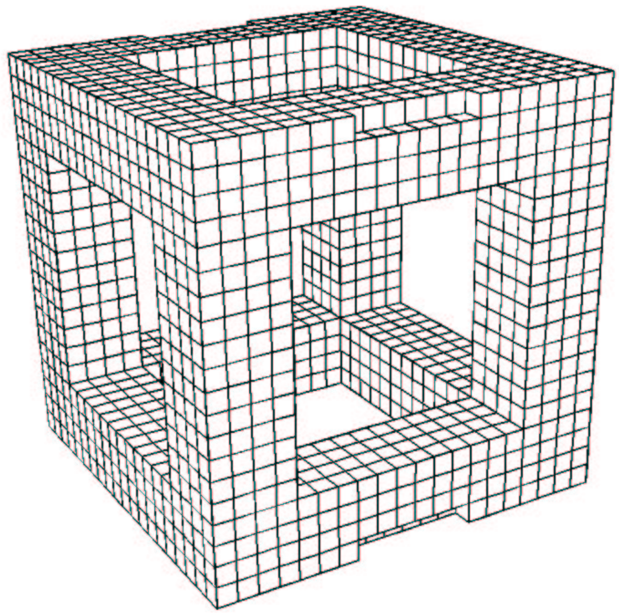


Figure 9.3: An example of a structured volume mesh[40].

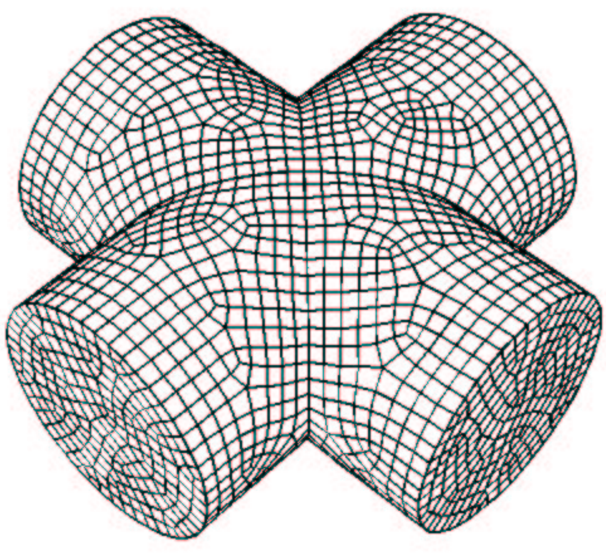


Figure 9.4: An example of an unstructured surface mesh[5].

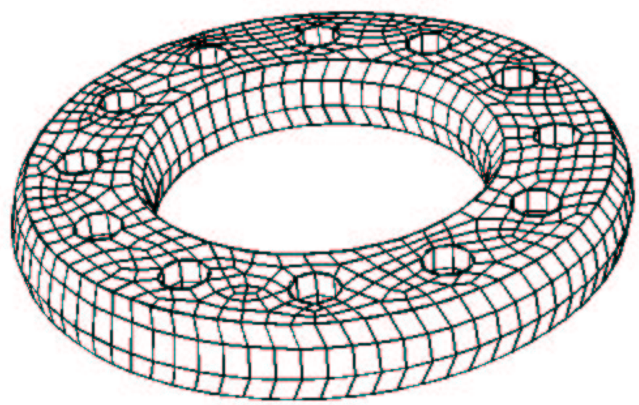


Figure 9.5: An example of a semi-structured volume mesh[18].

Bibliography

[1] Bank, R.E., and Xu, J. An algorithm for coarsening unstructured meshes. *Numerische Mathematik v 73* (1996), pp. 1–36.

[2] Benzley, S., Perry, E., Merkley, K., and Clark, B. A comparison of all-hexahedral and all-tetrahedral finite element meshes for elastic and elasto-plastic analysis. In *Proceedings, 4^{th} International Meshing Roundtable* (1995), Sandia National Laboratories, pp. 179–191.

[3] Blacker, T. Meeting the challenge for automated conformal hexahedral meshing. In *Proceedings, 9^{th} International Meshing Roundtable '00* (2000), Sandia National Laboratories, pp. 11-19.

[4] Blacker, T. The cooper tool. In *Proceedings, 5^{th} International Meshing Roundtable '96* (1996), Sandia National Laboratories, pp. 13–29.

[5] Blacker, T., Stephenson, M.B. Paving: A new approach to automated quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering 32* (1991), pp. 811-847.

[6] Borouchaki, H. and Frey, P. Optimization tools for adaptive surface meshing. *Trends in Unstructured Mesh Generation 220* (1997), pp. 81–88.

[7] Bussler, Michael L., and Ramesh, A. The eight-node hexahedral element FEA of part designs. *Foundry Management and Technology v 121 n 11* (1993), pp. 26–28.

[8] Cifuentes, A., and Kalbag, A. Performance study of tetrahedral and hexahedral elements in 3-d finite element structural analysis. *Finite Elements in Analysis and Design 12* (1992), pp. 313–318.

[9] De Cougny, H.L., and Shephard, M.S. Parallel refinement and coarsening of tetrahedral meshes. *International Journal for Numerical Methods in Engineering v 46* (1999), pp. 1101–1125.

[10] Frey, P.J. and Borouchaki, H. Unit surface mesh simplification. *Trends in Unstructured Mesh Generation 220* (1997), pp. 51–56.

[11] Frey, P.J and Borouchaki, H. Geometric evaluation of finite element surface meshes. *Geometric Evaluation of Finite Element Surface Meshes v 31* (1998), pp. 33–53.

[12] George, P. *Automatic Mesh Generation.* John Wiley & Sons, 1991.

[13] George, P.L., Hecht, F., and Saltel, E. Automatic mesh generator with specified boundary. *Computer Methods in Applied Mechanics and Engineering v 92* (1991), pp. 269–288.

[14] Hamann, B. A data reduction scheme for triangulated surfaces. *Computer Aided Geometric Design v 11* (1994), pp. 197–214.

[15] Hoppe, H. et. al. Mesh optimization. *Computer Graphics* Proc. ACM 1, Ann. Conf. Series, (1993), pp. 19–26.

[16] Jones, R.E., *QMESH: A Self-Organizing Mesh Generation Program* SLA - 73 - 1088, Sandia National Laboratories, (1974).

[17] Knupp, P. Algebraic mesh quality metrics for unstructured initial meshes. *Finite Elements in Design and Analysis v 39 n 3* (2003), pp. 217-241.

[18] KNUPP, P. Next-generation sweep tool: a method for generating all-hexahedral meshes on two-and-one-half dimensional geometries. In *Proceedings, 7th International Meshing Roundtable '98* (1998), Sandia National Laboratories, pp. 505–513.

[19] LO, S.H. Generating quadrilateral elements on plane and over curved surfaces. *Computers and Structures v 31 n 3* (1989), pp. 421-426.

[20] LU, Y. Feature based hex meshing methodology: Feature recognition and volume decomposition. *CAD Computer Aided Design v 33 n 3* (2001), pp. 221-232.

[21] MARECHAL, L. A new approach to octree-based hexahedral meshing. In *Proceedings, 10th International Meshing Roundtable '01* (2001), Sandia National Laboratories, pp. 209–221.

[22] MELANDER, D., TAUTGES, T., AND BENZLEY, S. Generation of multi-million element meshes for solid-based geometries: the dicer algorithm. *Trends in Unstructured Mesh Generation 220* (1997), 131–135.

[23] MESHKAT, S. AND TALMOR, D. Generating a mixed mesh of hexahedra, pentahedra and tetrahedra from an underlying tetrahedral mesh. *International Journal for Numerical Methods in Engineering v 49* (2000), pp. 17-30.

[24] MINGWU, L., AND BENZLEY, S. A multiple source and target sweeping method for generating all hexahedral finite element meshes. In *Proceedings, 5th International Meshing Roundtable '96* (1996), Sandia National Laboratories, pp. 217–225.

[25] MIYOSHI, K., AND BLACKER, T. Hexahedral mesh generation using multi-axis cooper algorithm. In *Proceedings, 9th International Meshing Roundtable '00* (2000), Sandia National Laboratories, pp. 89–97.

[26] MURDOCH, P. *The Spatial Twist Continuum: A Dual Representation of the All Hexahedral Finite Element Mesh.* PhD thesis, Brigham Young University, 1995.

[27] Ollivier-Gooch, C. Coarsening unstructured meshes by edge contraction. *International Journal for Numerical Methods in Engineering v 57 /* (2003), pp. 391-414.

[28] Owen, S. Constrained triangulation: application to hex-dominant mesh generation. In *Proceedings, 8th International Meshing Roundtable '99* (1999), Sandia National Laboratories, pp. 31–42.

[29] Owen, S. and White, D. Mesh-based geometry. *International Journal for Numerical Methods in Engineering v 58* (2003),pp. 375-395.

[30] Pelessonne, D. and Charman, C.M. Adaptive finite element procedure for non-linear structural analysis. *Developments in Pressure Vessels and Piping v 301* (1995),pp. 191–198

[31] Renze, K., and Oliver, J. Generalized unstructured decimation. *IEEE Computer Graphics and Applications v 16 n 6* (1996), pp. 24–32.

[32] Schneiders, R., Schindler, R., and Weiler, F. Octree-based generation of hexahedral element meshes. In *Proceedings, 5th International Meshing Roundtable '96* (1996), Sandia National Laboratories, pp. 205–215.

[33] Schroeder, W.J., Zarge, J.A., and Lorensen, W.E. Decimation of triangle meshes. *Computer Graphics v 26 n 2* Proc. ACM 1 (1992), pp. 65–70.

[34] Shepherd, J., Mitchell, S., Knupp, P., and White, D. Methods for multisweep automation. In *Proceedings, 9th International Meshing Roundtable '00* (2000), Sandia National Laboratories, pp. 77–87.

[35] Taniguchi, T. and Fillion, E. Numerical experiments for 3-dimensional flow analysis in a fracture rock with porous matrix. *Advances in Water Resources* Vol 39 No 2, (1996), pp. 97–107.

[36] Tautges, T., Blacker, T., and Mitchell, S. The whisker weaving algorithm: a connectivity-based method for all-hexahedral finite element meshes. *International Journal for Numerical Methods in Engineering 39* (1996), pp. 3327–3349.

[37] Turk, G. Re-tiling polygonal surfaces. *Computer Graphics v 26 n 2* (1992), pp. 55–64.

[38] Weingarten, V.I. The controversy over hex or tet meshing. *Machine Design v 66 n 8* (1994), pp. 74–78.

[39] Weiss, M. *Data Structures and Algorithm Analysis.* Benjamin/Cummings Publishing, Redwood City, CA, 1995.

[40] White, D. R. Automatic, quadrilateral and hexahedral meshing of pseudo-cartesian geometries using virtual subdivision. Master's thesis, Brigham Young University, 1996.

[41] Yamakawa, S., and Shimada, K. Hexhoop: Modular templates for converting a hex-dominant mesh to an all-hex mesh. In *Proceedings, 10^{th} International Meshing Roundtable '01* (2001), Sandia National Laboratories, pp. 235–246.